

# PATENT COOPERATION TREATY

## PCT

### NOTIFICATION OF ELECTION

(PCT Rule 61.2)

From the INTERNATIONAL BUREAU

To:

Assistant Commissioner for Patents  
United States Patent and Trademark  
Office  
Box PCT  
Washington, D.C.20231  
ETATS-UNIS D'AMERIQUE

in its capacity as elected Office

Date of mailing (day/month/year) 19 July 2000 (19.07.00)	
International application No. PCT/GB99/03891	Applicant's or agent's file reference JDC/5285399
International filing date (day/month/year) 23 November 1999 (23.11.99)	Priority date (day/month/year) 23 November 1998 (23.11.98)
Applicant BILCHEV, George	

1. The designated Office is hereby notified of its election made:

☒ in the demand filed with the International Preliminary Examining Authority on:

09 May 2000 (09.05.00)

☐ in a notice effecting later election filed with the International Bureau on:
2. The election ☒ was
☐ was not

made before the expiration of 19 months from the priority date or, where Rule 32 applies, within the time limit under Rule 32.2(b).

The International Bureau of WIPO 34, chemin des Colombettes 1211 Geneva 20, Switzerland  Facsimile No.: (41-22) 740.14.35	Authorized officer  Juan Cruz  Telephone No.: (41-22) 338.83.38
---	---

## PATENT COOPERATION TREATY

PCT

NOTIFICATION OF THE RECORDING  
OF A CHANGE(PCT Rule 92bis.1 and  
Administrative Instructions, Section 422)

From the INTERNATIONAL BUREAU

To:

BT GROUP LEGAL SERVICES  
Intellectual Property Department  
8th Floor, Holborn Centre  
120 Holborn  
London EC1N 2TE  
ROYAUME-UNI

Date of mailing (day/month/year) 09 August 2000 (09.08.00)	<b>IMPORTANT NOTIFICATION</b>
Applicant's or agent's file reference JDC/5285399	
International application No. PCT/GB99/03891	International filing date (day/month/year) 23 November 1999 (23.11.99)

## 1. The following indications appeared on record concerning:

☐ the applicant      ☐ the inventor      ☒ the agent      ☐ the common representative

Name and Address BERESFORD, Keith, Denis, Lewis Beresford & Co. 2-5 Warwick Court High Holborn London WC1R 5DJ United Kingdom	State of Nationality	State of Residence
	Telephone No. 0171 831 2290	
	Facsimile No. 0171 405 4092	
	Teleprinter No.	

## 2. The International Bureau hereby notifies the applicant that the following change has been recorded concerning:

☒ the person      ☒ the name      ☒ the address      ☐ the nationality      ☐ the residence

Name and Address BT GROUP LEGAL SERVICES Intellectual Property Department 8th Floor, Holborn Centre 120 Holborn London EC1N 2TE United Kingdom	State of Nationality	State of Residence
	Telephone No. 020-7492-8110	
	Facsimile No. 020-7242-0767	
	Teleprinter No.	

## 3. Further observations, if necessary:

## 4. A copy of this notification has been sent to:

<input checked="" type="checkbox"/> the receiving Office	<input type="checkbox"/> the designated Offices concerned
<input type="checkbox"/> the International Searching Authority	<input checked="" type="checkbox"/> the elected Offices concerned
<input checked="" type="checkbox"/> the International Preliminary Examining Authority	<input type="checkbox"/> other:

The International Bureau of WIPO 34, chemin des Colombettes 1211 Geneva 20, Switzerland	Authorized officer Dominique DELMAS
Facsimile No.: (41-22) 740.14.35	Telephone No.: (41-22) 338.83.38

# PATENT COOPERATION TREATY

# PCT

## INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference <b>JDC/5285399</b>	<b>FOR FURTHER ACTION</b>		see Notification of Transmittal of International Search Report (Form PCT/ISA/220) as well as, where applicable, item 5 below.
International application No. <b>PCT/GB 99/ 03891</b>	International filing date (day/month/year) <b>23/11/1999</b>	(Earliest) Priority Date (day/month/year) <b>23/11/1998</b>	

Applicant

**BRITISH TELECOMMUNICATIONS PUBLIC LIMITED .. et al**

This International Search Report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This International Search Report consists of a total of 3 sheets.

☒ It is also accompanied by a copy of each prior art document cited in this report.

**1. Basis of the report**

a. With regard to the language, the international search was carried out on the basis of the international application in the language in which it was filed, unless otherwise indicated under this item.

☐ the international search was carried out on the basis of a translation of the international application furnished to this Authority (Rule 23.1(b)).

b. With regard to any nucleotide and/or amino acid sequence disclosed in the international application, the international search was carried out on the basis of the sequence listing :

☐ contained in the international application in written form.

☐ filed together with the international application in computer readable form.

☐ furnished subsequently to this Authority in written form.

☐ furnished subsequently to this Authority in computer readable form.

☐ the statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

☐ the statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished

2. ☐ Certain claims were found unsearchable (See Box I).

3. ☐ Unity of invention is lacking (see Box II).

**4. With regard to the title,**

☒ the text is approved as submitted by the applicant.

☐ the text has been established by this Authority to read as follows:

**5. With regard to the abstract,**

☒ the text is approved as submitted by the applicant.

☐ the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box III. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

**6. The figure of the drawings to be published with the abstract is Figure No.**

☒ as suggested by the applicant.

☐ because the applicant failed to suggest a figure.

☐ because this figure better characterizes the invention.

3  
☐ None of the figures.

## INTERNATIONAL SEARCH REPORT

International Application No.

PCT/GB 99/03891

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 H04L9/06 H04L9/00

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>NANDI S ET AL: "THEORY AND APPLICATIONS OF CELLULAR AUTOMATA IN CRYPTOGRAPHY" IEEE TRANSACTIONS ON COMPUTERS, NEW YORK (US), vol. 43, no. 12, 1 December 1994 (1994-12-01), pages 1346-1356, XP000484159 abstract page 1348, right-hand column, line 6 - line 34 page 1351, right-hand column, last paragraph -page 1352, left-hand column, line 16</p> <p style="text-align: center;">— — — — — — / —</p>	1, 3, 12, 23, 25, 41, 49, 53, 63, 71

☒ Further documents are listed in the continuation of box C.☒ Patent family members are listed in annex.

## \* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&amp;" document member of the same patent family

Date of the actual completion of the international search

20 March 2000

Date of mailing of the international search report

27/03/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Holper, G

## INTERNATIONAL SEARCH REPORT

International Application No

PCT/GB 99/03891

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 89 07375 A (MOTOROLA INC) 10 August 1989 (1989-08-10)  page 4, line 1 -page 5, line 30 page 6, line 1 - line 18	1, 12, 13, 23, 34, 41, 53, 54, 63, 75
A	EP 0 267 647 A (PHILIPS) 18 May 1988 (1988-05-18) column 2, last line -column 3, line 13 column 4, line 1 - line 45 column 5, line 1 - line 28	1, 23, 41, 63
A	PATENT ABSTRACTS OF JAPAN vol. 017, no. 456 (E-1418), 20 August 1993 (1993-08-20) & JP 05 102960 A (NEC CORP), 23 April 1993 (1993-04-23) abstract	7-10, 28-31, 45-48, 67-70
A	US 4 984 271 A (GOTO) 8 January 1991 (1991-01-08) abstract; figure 3	1, 23, 41

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

P 99/03891

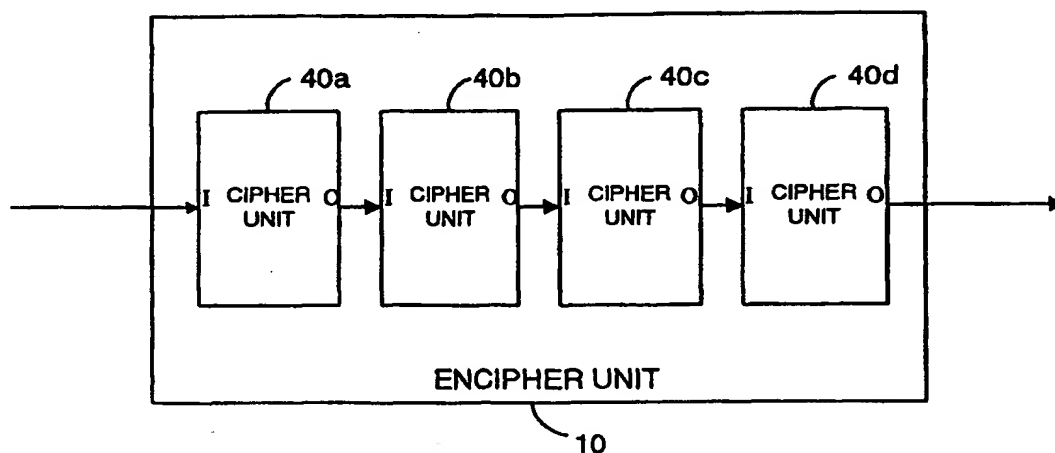
Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 8907375	A	10-08-1989	US 4914697 A AT 139392 T CA 1336721 A DE 68926670 D DE 68926670 T EP 0398931 A HK 1004585 A JP 3500117 T KR 9614682 B	03-04-1990 15-06-1996 15-08-1995 18-07-1996 19-12-1996 28-11-1990 27-11-1998 10-01-1991 19-10-1996
EP 267647	A	18-05-1988	NL 8602847 A AU 611653 B AU 8095087 A CA 1291801 A JP 2628660 B JP 63135035 A US 4890324 A	01-06-1988 20-06-1991 12-05-1988 05-11-1991 09-07-1997 07-06-1988 26-12-1989
JP 05102960	A	23-04-1993	NONE	
US 4984271	A	08-01-1991	JP 63278438 A	16-11-1988



## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>7</sup> : <b>H04L 9/06, 9/00</b>		A1	(11) International Publication Number: <b>WO 00/31916</b>
			(43) International Publication Date: 2 June 2000 (02.06.00)
(21) International Application Number: PCT/GB99/03891		(81) Designated States: CA, SG, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).	
(22) International Filing Date: 23 November 1999 (23.11.99)			
(30) Priority Data: 9825644.9 23 November 1998 (23.11.98) GB		<b>Published</b> <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>	
(71) Applicant (for all designated States except US): BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).			
(72) Inventor; and (75) Inventor/Applicant (for US only): BILCHEV, George [BG/GB]; 33 Elmers Lane, Ipswich IP5 2GW (GB).			
(74) Agents: BERESFORD, Keith, Denis, Lewis et al.; Beresford & Co., 2-5 Warwick Court, High Holborn, London WC1R 5DJ (GB).			

(54) Title: A CIPHER



## (57) Abstract

A cipher is disclosed for enciphering and deciphering a signal which comprises a plurality of sequentially coupled cipher units, each cipher unit being operable to carry out a reversible operation on the signal. The couplings between cipher units can be randomly configured using a cipher code. The cipher code can be secretly shared between the encipher and decipher. A signal which is enciphered using this technique is thus deciphered using a randomly selected cipher circuit as described by the cipher code.

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakistan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		



### A CIPHER

The present invention generally relates to a cipher and in particular to a cipher in which the secret cipher code which is required for both enciphering and  
5 deciphering is information which describes the process used to carry out enciphering.

Two commonly used types of cryptographic algorithms are private key algorithms which use a single shared key and public key algorithms which use two keys: a public  
10 key and a private key.

In these prior art algorithms the encryption process used is fixed although the particular encryption process can be selectable by user e.g. by using a particular encryption program (algorithm). The security of the  
15 encryption is provided by the key which is secretly exchanged between the encrypter operator and decrypter operator.

Such currently implemented ciphers are not easily scalable since they are defined for a specific block size and key size. In many instances the key is not big  
20 enough e.g. many ciphers have only 64 bit keys.

In accordance with a first aspect, the present invention provides an encipher apparatus for enciphering a signal that comprises a plurality of sequentially  
25 implemented cipher units, wherein each implemented cipher unit carries out an operation on the signals being ciphered which it can reverse. The apparatus also includes means for configuring the couplings between the implemented cipher units.

Another aspect of the present invention provides an encipher method wherein couplings between cipher modules are configured and the signal to be enciphered is sequentially passed through the configured cipher units.

5 Each cipher unit is operative to carry out a process on the signal which it is capable of reversing.

Thus the present invention provides a universal cipher which is capable of implementing any cipher process. The encryption which is carried out on the  
10 signal is dependent upon the coupling configuration of the cipher units. The couplings are configurable without changing the configuration of the cipher units. This configuration is freely selectable and is preferably selected randomly or psuedo-randomly and automatically  
15 for the usual security reasons to prevent any element of predictability.

In one embodiment the cipher units are identical and thus perform identical reversible operations. The invention does however encompass the use of a plurality  
20 of types of cipher units wherein the sequential pattern of the different types is information that must be shared to allow deciphering of the signal encipher using the pattern.

The cipher units can be implemented in many  
25 different ways such as a reversible circuit either implemented in logic gates or in logic steps performed by a computer, analog circuitry, or optical elements. In fact, the encipher units can be implemented by any physical process which is reversible.

A further aspect of the present invention provides decipher apparatus for deciphering an enciphered signal and comprising a plurality of sequentially implemented cipher units wherein each cipher unit is implemented to  
5 carry out a process which it can reverse. The apparatus also includes means for configuring couplings between the cipher units dependent upon the enciphering of the enciphered signal wherein the couplings are configurable without changing the configuration of the cipher units.

10 Another aspect of the present invention provides a decipher method wherein couplings between a plurality of cipher units are configured and an enciphered signal is passed through the cipher units. Each of the cipher units carries out an operation on the enciphered signal  
15 which it is capable of reversing.

In one embodiment the cipher units are identical and carry out identical operations on the enciphered signal.

The use of reversible cipher units in both the encipher and decipher enable the configuration of the  
20 units to be the same although the implementation will be reversed. It is this reversibility which allows the use of information describing the cipher process to be used as a secret cipher code between the encipher and decipher. In other words, instead of a secret key to be  
25 shared by the sender and receiver of an encrypted message, a cipher code which describes the cipher process is shared instead.

Thus the invention is similar to the conventional symmetric cryptography technique except that there is no

single shared key but instead a single shared cipher code containing information describing the cipher process.

In a similar manner to use of a private key, the cipher code can be determined by either party in a two-party communication of an encrypted signal. In other words, either the recipient of an encrypted signal can request for the cipher code to be used and secretly pass this to the party for use in transmitting the encrypted signal, or the party transmitting the encrypted signal can secretly inform the recipient of the cipher code to be used to decrypt the signal.

The invention is equally applicable to the encryption of a signal which is not transmitted and which is instead stored securely e.g. the encryption and storage of data in a computer to prevent unauthorised access. In this example there need only be one party.

Because the cipher modules are reversible, the encipher and decipher can comprise the same apparatus. Thus, for duplex communication of an encrypted signal or for the storage of encrypted data for retrieval and decryption it is possible for the same cipher module to be used but in reverse order for deciphering.

In one embodiment each cipher unit comprises a controllable switch module having a plurality of inputs at least one of which acts to control a switching operation. Such a switch module can be implemented as a reversible gate such as a Fredkin's gate or an AND/NAND gate. Such gates can be implemented in logic either as

logic gates such as a programmable logic circuit or as logic steps implementing the gates in a computer program.

An advantage of this invention is that it is inherently scalable since the number of cipher modules  
5 can be varied dependent upon the configuration. Further, the size of the data block can be varied. This will also depend upon the configuration of the cipher unit.

Conveniently, ciphers are usually implemented digitally as a computer program. The programmability of  
10 a general purpose computer provides the facility for a universal cipher. Since a computer is capable of implementing a reversible computational process which implements a one-to-one mapping, and since general purpose computers are available which can be programmed  
15 to carry out any reversible computational process, any reversible computational process can be implemented thus implementing any one-to-one mapping. Any general purpose reversible computer can be used as a universal cipher. The use of a computer program to implement the reversible  
20 process further enables a user to select the type of reversible process to be implemented.

Since the present invention can be implemented on a general purpose computer by a suitable program, the present invention can be embodied as a storage medium  
25 storing instructions for controlling a processor e.g. a floppy disc, CD-ROM, smartcard, and programmable memory. Further, since the computer program can be transmitted over a network to be received and implemented on a computer, the present invention can be embodied as a

signal carrying the processor implementable instructions.

Embodiments of the present invention will now be  
5 described with reference to the accompanying drawings,  
in which:

Figure 1 is a schematic illustration of a cipher  
system.

Figure 2 is a schematic illustration of the  
10 reversibility of the cipher.

Figure 3 is a schematic illustration of the encipher  
unit.

Figure 4 is a schematic illustration of the decipher  
unit.

15 Figure 5 is a schematic illustration of a Fredkin's  
gate.

Figure 6 is a schematic illustration of the cipher  
code structure describing the configuration of Fredkin's  
gate.

20 Figure 7 is a schematic illustration of a gate  
circuit using Fredkin's gates.

Figure 8 is an illustration of the cipher code for  
the circuit of Figure 7.

Figure 9 is a functional diagram of a cipher code  
25 generator.

Figure 10 is a functional diagram of an encipher.

Figure 11 is a functional diagram of a decipher.

Figure 12 is a diagram of use of the cipher in the  
transmission of encrypted data.

Figure 13 is a diagram of a specific embodiment wherein the cipher code and possibly the cipher circuit is exchanged using a smartcard.

Figure 14 is a diagram of a processing apparatus  
5 capable of implementing the cipher.

Figure 15 is a flow diagram showing the generation and exchange of encrypted data using the cipher.

Figure 16a is a flow diagram illustrating the generation of the cipher code.

10 Figure 16b is a schematic diagram of the cipher code.

Figure 17 is a flow diagram illustrating the encipher process.

15 Figure 18 is a flow diagram illustrating the decipher process.

Figure 19 is a Fredkin's gate illustrated as a three-input logic gate.

Figure 20 is an implementation of the logic gate of Figure 19 using AND, OR and NOT gates.

20 Figure 21 is a diagram of an implementation of the Fredkin's gate using multiplexors.

Figure 22 is a diagram of an implementation of the Fredkin's gate using three-state buses.

25 Figure 23 is a diagram of an AND/NAND gate as an alternative reversible gate to the Fredkin's gate.

Referring now to the drawings, Figure 1 illustrates a cipher system in general wherein an encipher unit 10 generates an enciphered or encrypted signal using shared

configuration information. The configuration information is used to configure the encipher unit 10. The encrypted signal is then transmitted by a transmission medium 20 to a recipient decipher unit 30 which also has the shared configuration information. The decipher unit 30 is configured in accordance with the configuration information and operates the reverse of the process carried out by the encipher unit 10 to thereby decipher the signal.

10        Although in this embodiment a transmission medium 20 is illustrated, the transmission medium could simply comprise a storage medium on which the encrypted data is stored. Thus the operator generating the encrypted signal and the operator receiving the encrypted signal  
15        may in fact be the same.

Figure 2 schematically illustrates the reversibility of a cipher unit to act either as a decipher unit 30 or an encipher unit 10.

Figure 3 illustrates in more detail the cipher unit  
20        10 which is comprised of cipher units 40a to 40d. Although in this embodiment four cipher units are illustrated, in a practical embodiment this would typically be at least four times the block size e.g. for a block size of 128 bits the number of cipher units is  
25        at least 512. The number will however depend on the level of security desired. As can be seen in Figure 3 the input signal is received at the input I of each of the cipher units 40a to 40d sequentially.



Figure 4 illustrates a decipher unit 30 in more detail. The decipher unit comprises the same configuration of cipher units 40a to 40d as in the encipher unit 10. However, in order to decipher the enciphered signal, it is passed in reverse through the cipher units 40a to 40d.

A specific implementation of the cipher will now be described in which the cipher unit is implemented using Fredkin's gates. Such a gate is illustrated in Figure 5. It is well known that a Fredkin's gate is both reversible (i.e. circuits implemented by Fredkin's gates can be run backwards to uncompute) and universal (i.e. can be used to design circuits that implement all one-to-one mappings).

In the Fredkin's gate as illustrated in Figure 5, the input A is used to control the exchange of data on inputs B and C. Thus Fredkin's gate performs a controlled exchange operation. If  $A=1$ , B and C are not exchanged i.e.  $B'=B$  and  $C'=C$ . If however,  $A=0$ ,  $B'=C$  and  $C'=B$ . In mathematical notation  $B'=AB+\bar{A}C$  and  $C'=\bar{A}B+AC$ .

Fredkin's gate is a conservative logic gate i.e. it preserves the numbers of 0's and 1's from the input to the output. In a cipher system this is undesirable, thus in order to break the conservation, NOT gates are selectively applied to the outputs to invert them. The selective inversion are operations which are inherently reversible and thus do not affect the reversibility of the circuit.

Having selected the type of reversible circuit used as the cipher unit, it is then necessary to determine a cipher code to describe the arrangement of the circuits. Figure 6 illustrates once such method wherein each Fredkin's gate is described by a four segment code. Each of the first three segments describe pin numbers to which the gates are attached in the input signal. The last segment is used to encode a description of the presence or the absence of inverters on each of the output pins A', B' and C'.

Consider an encipher process in which it is decided that the input signal is to be enciphered in 8 bit blocks. The input data thus comprises a 8 bit array indexed from 000 to 111. Each segment of the cipher code for each gate thus comprises a 3 bit code. For example the sequence 010 111 110 110 defines a gate with A of the gate attached to pin 2 (010) of the input, B attached to pin 7 (111) of the input and C attached to pin 6 (110) of the input. The last segment defines that output A' and B' are passed through NOT gates i.e. inverted. The 8 bit signal as modified by the first gate is then used as an input to the second gate and so on.

Figure 7 illustrates schematically an arrangement of 10 cipher circuits comprised of Fredkin's gates and NOT gates and Figure 8 illustrates the cipher code used to describe the circuit.

As can clearly be seen the cipher code simply comprises a digital code. The digital code is defined in as  $(3 \times \log_2 N) + 3$  bit blocks and each block defines a

cipher unit where  $N$  is the number of input bits i.e. the block size. The total number of bits to define a circuit is  $M((3 \times \log_2 N) + 3)$  where  $M$  is the number of cipher units. Whilst it is possible to allow a user to select a code freely by for example choosing a "password" in ASCII code which can be translated to binary (e.g. for the 8 bit input, 10 gate example in Figure 7, a 15 character 8 bit ASCII password could be used to describe the circuit), it is preferable for the usual security reasons to randomly generate a code which describes a random configuration of the gates.

In this example, in order to encrypt the signal it is passed from left to right through the gates. In order to decrypt the signal it is passed from right to left. Thus in decryption as the signal is input into each Fredkin's gate the mask defines whether it is to be first inverted before being switched in accordance with the value of the input on pin A.

It is possible for some of the gates to be implemented in parallel so long as their outputs and inputs are not coincident.

Figure 9 is a functional diagram of a cipher code generating apparatus. A random number generator 100 generates a random number to be used to form the cipher code. This is input through a validity checker 110 which checks whether the random number is valid. For example, the random number cannot result in two pins or more of the gate being on the same input line i.e. it defines a

valid gate. The validity checker 30 requires information on the block size in order to do this check.

The random number is then input into the cipher code (circuit array) forming unit 120 in order to build the cipher code describing the circuit array. The cipher code forming unit 120 also receives an input from the cipher code parameter selector 130 which is operable by a user to select the number of bits or cipher units to be implemented in the cipher, and to select the data block size. Also, in a general purpose computer, it is possible to select the type of reversible gates to be used. Since there is however a limited number of possible types of gates currently known which can be implemented reversibly, allowing such a selection does not greatly increase the level of security at present.

Once the cipher code has been formed it is then stored in a non-volatile memory 140 for use in enciphering and deciphering data. If data is to be transmitted between two parties, the cipher code must be secretly shared. Where there has been selection of parameters in building the cipher code i.e. the number of gates, the block size and the type of gates, this information will also need to be shared so that the cipher code can be used properly to implement a cipher circuit for both encryption and decryption.

Figure 10 is a functional diagram of an encipher apparatus. A signal to be enciphered is input by the data input device 200. This is then passed to a data block former 210 which forms the input signal into blocks

of data which can be sequentially passed through the encipher apparatus. The block of data is then passed into the working memory 220 as an array of N bits where N is the block size. A circuit implementor 250 then  
5 implements the cipher circuit in accordance with the circuit array stored in the non-volatile memory 260. The circuit array comprises a M array of gate descriptions, where each gate description comprises four segments (as illustrated in Figure 6). The circuit implementor 250  
10 will operate on the data block in the working memory 220 to implement each of the gates sequentially. Circuit implementor 250 will therefore control the data block former 410 to input a block of data into the working memory 220 when it is ready to operate on it. Once the  
15 enciphering operation has been completed on the data block in the working memory 220, the circuit implementor 250 controls the passage of the data block from the working memory 220 into a memory 230. The encipher data block can then be passed out block-by-block into a data  
20 output device 240 which can either output the encipher data block block-by-block or can wait until all of the data blocks enciphered and output the complete enciphered signal.

Figure 11 is a functional diagram of the decipher  
25 apparatus in accordance with an embodiment of the present invention. Enciphered data is received by the encipher data input device 300 and is formed into enciphered data blocks by the enciphered data block former 310. The passage of enciphered data blocks into a working memory

320 is then controlled by a reverse circuit implementor 350. When a data block is in the working memory 320 the reverse circuit implementor 350 implements the encipher circuit in accordance with the circuit array stored in the non-volatile memory 260 in reverse. Once all of the gates of the circuit array have been implemented in reverse and thus the enciphered data block has been deciphered, it is output into the memory 330 under the control of the reverse circuit implementor 350. The decipher data block can then be output to the output device 340 which can then either output each of the data blocks sequentially or wait until the complete signal has been deciphered before outputting it.

Figure 12 illustrates an application of the cipher for the communication of enciphered data between computers 50, 51 and 52.

Computer 50 implements the encipher and generates enciphered data. This can either be stored on a non-volatile memory device such as a floppy disc 54 and passed to another computer 51 for deciphering, or it can be broadcast or transmitted over a network 53 for reception by computer 52 for deciphering there. Before the exchange of enciphered data however, it is necessary for the operators of computers 50 and 51 or 50 and 52 to secretly exchange the cipher code. This can be done by any conventional secret means such as a secure telephone call, a secure facsimile transmission or by letter, by courier or even by a secure e-mail.

Figure 13 illustrates another embodiment of the present invention wherein a computer 60 is provided with a smartcard programmer/reader 61. In this embodiment it is possible for a smartcard to be loaded with the decipher program as well as the cipher code. The smartcard can then be given to the intended recipient of enciphered data. Thus the intended recipient of the enciphered data can simply insert the smartcard into a smartcard reader and the processor on the smartcard will implement the deciphering circuit and thus inherently the enciphering circuit. Thus the smartcard can be used for both transmitting and receiving enciphered data. This embodiment can be used by an institution such as a financial institution (Alice). A user (Bob) will be issued with a smartcard and will be able to communicate securely with the institution by inserting the smartcard into the reader 61 e.g. an automatic teller machine (ATM).

Figure 14 is a schematic diagram of the implementation of the cipher in a general purpose computer. The computer is provided with a bus 79 to communicate between operational units. A modem 70 is provided for connection over a telecommunications line 78 to transmit and receive enciphered data. Also a network card 80 is provided for connection over a network to transmit and receive data. A keyboard 74 is provided for inputting data and a display 71 is provided for displaying deciphered data. A processor 72 implements the encipher circuit either in a forward direction for

enciphering or in a reverse direction for deciphering in accordance with the circuit array stored in the memory section 77. The processor 72 operates in accordance with the circuit emulation program stored in the program memory 75. During the operation of the processor 72 data is temporarily stored in the working memory 76 and at the end of the enciphering or deciphering process the enciphered or deciphered data can be stored in the data storage device 73 which can comprise non-volatile storage media such as a floppy disc, a hard disc, a writable CD-ROM, or EPROM.

The method of operation of the cipher of this embodiment of the present invention will now be described with reference to Figures 15 to 18.

Figure 15 illustrates the steps involved in generation of the cipher code, enciphering of data, the transmission of the enciphered data and the deciphering of the enciphered data.

In step S1 a type of reversible processing is predetermined or selected e.g. Fredkin's gates. In step S2 the number of gates M and the block size N of the data is selected. In step S3 the cipher code (or circuit array) is then generated. In step S4 the cipher code is exchange secretly between Alice and Bob. In step S5 Alice enciphers data using a circuit configured according to the circuit array. Alice then communicates the enciphered data to Bob in step S6. In step S7 Bob decipheres the ciphered data using a reverse circuit configured according to the circuit array (cipher code).



Figure 16a illustrates in more detail the steps involved in a generation of the cipher code.

In step S10 a variable  $m$  is set to 0. This variable acts as the gate number. In step S11 a random number having  $P$  bits is then generated, where  $P$  is the number of bits necessary to describe a gate. In the example given hereinabove using Fredkin's gates,  $P=12$  (4 segments each of 3 bits). In step S12 a check is carried out to determine whether this is a valid random number. One of the tests is whether the random number defines the gate having two or more pins on the same input data address which is not allowed. In step S13 if the random number is valid the gate number  $m$  is incremented and in step S14 the generated random number is stored indexed by  $m$ . In step S15 it is then determined whether random numbers have been generated for all of the gates i.e.  $m=M$ . If not, the process returns to step S11 for the generation of further random numbers. If random numbers have been generated defining all of the gates then the process ends in step S16.

Figure 16b illustrates the circuit array which comprises a  $P \times M$  matrix. The matrix is indexed by  $M$  where each entry comprises  $P$  bits divided into four segments A, B, C and M each of 3 bits.

The process of enciphering data will now be described with reference to Figure 17.

In step S20 the data to be enciphered is input and the prestored circuit array (secret cipher code) indexed by  $m$  is read. The first  $N$  bits of data are then read as

a data block. If there are less than N bits of data, padding data is generated in order to make up N bits. The N bits of data are then loaded into the working array in step S22 and in step S23 the gate counter m is set to 1. In step S24 the first segment A for gate m in the circuit array is read and this is used to address a data bit from the working array in step S25. In step S26 it is then determined whether the read data bit is 0. If it is not 0 then there is no exchange of data between input B and C and the process proceeds to step S30. If it is 0 then data bits B and C are exchanged. Thus in step S27 the second and third segments B and C for the gate m in the circuit array is then read and these are used to address two data bits in the working array. In step S29 these data bits are then exchanged and the process proceeds to step S30 where a mask bit counter b is set to 1 to index the first mask bit for segment A.

In step S31 the  $b^{\text{th}}$  bit of the mask is read and in step S32 it is determined whether this is zero. If it is not zero the data bit in the working array addressed by the  $b^{\text{th}}$  segment is inverted in step S33 otherwise no action is taken. In the next step S34 is determined whether all of the mask bits have been read i.e.  $b=3$  indicating that the mask bit for segment C has been read. If not the mask bit counter b is incremented in step S35 to index the next mask bit for segment B or C and the process returns to step S31. If all of the mask bits have been read it is then determined whether all of the gates have been implemented i.e.  $m=M$  in step S36. If not,

the gate counter  $m$  is incremented and the process returns to step S24. Otherwise in step S38 the working array is output as a block of enciphered data. In step S39 it is then determined whether the data has all been enciphered and if not, in step S40 the next  $N$  bits of data are input and padded if necessary and the process returns to step S22. Otherwise the process ends in step S41 since all of the data has been enciphered.

The process of deciphering enciphered data will now be described with reference to Figure 18.

In step S50 the enciphered data is input and the prestored circuit array indexed by  $m$  is read. The first  $N$  bits of enciphered data are then read in step S51. The  $N$  bits of enciphered data are then loaded into the working array in step S52 and in step S53 the gate counter  $m$  is set equal to  $M$  i.e. the first gate to be implemented is in fact the last gate in the array so that the gates are implemented sequentially in reverse. In step S54 the mask bit counter  $b$  is then set to the first mask bit for segment A and in step S55 the  $b^{\text{th}}$  bit of the mask is read. It is then checked in step S56 whether this is zero and if not the data bit in the working array addressed by the  $b^{\text{th}}$  segment is inverted in step S57 otherwise no action is taken. The process then proceeds to step S58 wherein it is determined whether all of the mask bits have been read i.e.  $b=3$ . If not, in step S59 the mask bit counter  $b$  is incremented to index the next mask bit for segment B or C and the process returns to step S55. If all of the mask bits have been read for the

mask segment, in step S60 the first segment A for gate  
m in the circuit array is read. A data bit in the  
working array addressed by this first segment A is then  
read in step S61 and it is determined whether this is  
5 zero in step S62. If it is zero the second and third  
segment B and C for gate m in the circuit array are read  
in step S63 and in step S64 the data bits in the working  
array addressed by the second and third segments B and  
C are read. These are then exchanged in step S65 and the  
10 process proceeds to step S66. If in step S62 the data bit  
addressed by the first segment A is not zero the process  
proceeds to step S66. In step S66 it is determined  
whether the process has just been carried out for the  
first gate i.e.  $m=1$  indicating that the deciphering of  
15 the current block has finished. If not in step S67 the  
gate counter is decremented and the process returns to  
step S54, and if so in step S68 the working array is  
output as a deciphered data block. In step S69 it is  
determined whether all of the gates have been deciphered  
20 and if not in step S70 the next N bits of deciphered data  
are read. The process then returns to step S52. If in  
step S69 it is determined that all of the data has been  
deciphered, in step S71 any data corresponding to padding  
data in the last block is ignored and the process ends  
25 in step S72.

In the above enciphering and deciphering embodiment  
each gate is implemented in software sequentially.

In the embodiment described hereinabove the  
reversible circuit is implemented by Fredkin's gates. The

Fredkin's gate can be viewed as a three-input, three-output logic gate as illustrated in Figure 19. This can be implemented using AND, OR and NOT logic gates (which are not reversible) as illustrated in Figure 20.

5 Of course, since the logic gates can only conduct signals one-way in order for the circuit to be reversible, it must perform an operation which is symmetric i.e. if the output of the circuit is put back as an input, the original input will be obtained. This is because a  
10 Fredkin's gate is an inverse of itself.

Thus the circuit illustrated in Figure 20 can comprise one of the cipher units 40a to 40d illustrated in Figures 3 and 4.

Another implementation of the Fredkin's gate can be  
15 chosed using multiplexer as illustrated in Figure 21. each multiplexer 400 and 401 receives two input signals and one control signal. If the control signal is zero then the first input signal is passed. If the control signal is one the second input signal is passed.

20 The Fredkin's gate can also be implemented by three-state buses as illustrated in Figure 22.

All of the three circuits given hereinabove can either be implemented in software using a computer program which generates the circuits and simulates them  
25 or using electronic hardwired circuits. It is thus possible for Alice and Bob to be supplied with off the shelf programmable gate array chips and with the software that downloads the cipher circuit description onto the chip. Such software languages for circuit descriptions

can for example comprise Verilog-HDL. In order for Alice and Bob to establish the secret communications, the downloading of the cipher circuit description will only be done once. When the circuit is implemented using  
5   hardwired circuits, Alice and Bob can use one circuit for encryption and one circuit for decryption. It is however possible to use only one circuit by downloading the circuit description at the time when communications take place. For example, if Alice wishes to send an encrypted  
10   message she downloads the cipher code (circuit array) onto the chip. If she receives a message she can download the decryption circuit onto the chip.

In the embodiment given above, Fredkin's gate is implemented in logic. However, Fredkin's gate can be  
15   implemented in many different ways, for example, it is possible to implement the Fredkin's gate in optics. The device which can be used to implement the Fredkin's gate in optics is the Mach/Zehnder interferometer switch. Such a switch is disclosed in a paper by J. Dommelly et al  
20   entitled "A Gallium Arsenide Electro-optical Interferometer Modulator", (Proc. 7th Topical Meeting on Integrated and Guided Wave Optics, Kissimmee 1984), the content of which is hereby incorporated in full by reference.

25   Although in the above embodiments, the use of Fredkin's gate has been described, the cipher units of the present invention can be implemented in many different ways. For example, another form of reversible universal logic is the AND/NAND gate (which is also known

as Toffoli's gate). The operation of the AND/NAND gate can be given by:

$$\begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_{n-1} \\ x_n \end{pmatrix} \rightarrow \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_{n-1} \\ x_n \oplus x_1 x_2 \dots x_{n-1} \end{pmatrix}$$

5        The AND/NAND gate is illustrated in Figure 23. In this gate the input on (n-1) of the n inputs act to switch the n<sup>th</sup> input by virtue of an AND gate receiving the (n-1) inputs and acting on an XOR gate on the n<sup>th</sup> input. Details on this particular type of gate are given  
10 in the paper by T. Toffoli entitled "Bicontinuous Extensions of Invertible Combinatorial Functions" (Mathematical Systems Theory, Vol. 14, pp. 13-23) the content of which is hereby incorporated by reference.

15        The AND/NAND gate can be implemented not just in logic as illustrated in Figure 23, but by any physical system.

20        In a system for implementing the cipher units of the present invention, any reversible computational system can be used and the present invention is not limited to the use of circuit implementation. For example, reversible cellular automata can be used as described in "Computation and Construction Universality of Reversible Cellular Automata" by T. Toffoli (J. Comput. Sys. Sci.,

Vol. 15, 1977, pp. 213-231), the content of which is hereby incorporated in full by reference, a reversible Turing machine as described in "Logical Reversibility of Computation" by C. Bennett (IBM J. Res. Dev. 6, 1973 pp. 525-532), the content of which is hereby incorporated in full by reference, quantum computing, for a "billiard ball", model of computation as described in "Conservative Logic" by E. Fredkin and T. Toffoli (International Journal of Theoretical Physics, Vol 21. nos. 3/4, 1982) the content of which is hereby incorporated in full by reference, for example.

In the embodiments described hereinabove, for security, a random number generator is used in order to randomly generate a circuit configuration. The random number generator is not essential to the present invention but does increase the level of security. Any of the standard strong real number generators available for crypto-software libraries can be used, or a true physical random source can be used. The generation of random or pseudo-random numbers is well known in the art.

It will be apparent to the skilled person in the art that the present invention can be implemented by providing Alice and Bob with a program that generates random circuits and simulates them. Generally in the software implementation the circuits will only be generated and simulated using the cipher code (circuit array) when the signal is input to be enciphered or deciphered.



If the cipher is to be implemented using programmable hardware, a manufacturer will provide Alice and Bob with a conventional programmable gate array and logic to set it up to run as a cipher.

5        Although the present invention has been described hereinabove with reference to specific embodiments, it will be apparent to the skilled person in the art that the present invention is not limited to the specific embodiments and modifications will be apparent to the  
10 skilled person in the art within the spirit and scope of the present invention.

**CLAIMS:**

1. Encipher apparatus for enciphering a signal, comprising:

5 a plurality of encipher functional modules sequentially coupled to operate sequentially on the signal, each encipher functional module having a plurality of inputs and a plurality of outputs and being operable to carry out a reversible operation on the  
10 signal input to said inputs; and

configuring means for configuring couplings between the plurality of outputs and inputs between said encipher functional modules.

15 2. Encipher apparatus according to claim 1, wherein said encipher functional modules are of a single type.

3. Encipher apparatus according to claim 1 or claim 2, wherein each said encipher functional module comprises  
20 a controllable switch module, and at least one of said inputs acts to control a switching operation.

4. Encipher apparatus according to any preceding claim wherein each of said encipher functional modules comprises a reversible gate.

5 5. Encipher apparatus according to claim 3, wherein said reversible gate comprises a Fredkin's gate or an AND/NAND gate.

6. Encipher apparatus according to any preceding claim  
10 wherein said configuring means is operative to configure the couplings between said encipher functional modules in accordance with information describing the configuration of the couplings between said encipher functional modules.

15

7. Encipher apparatus according to claim 6, including means for receiving said information.

8. Encipher apparatus according to claim 6, including  
20 means for generating said information.

9. Encipher apparatus according to claim 8, wherein the generating means includes a random or pseudo-random number generator and is operative to use random or pseudo-random numbers generated by said random or pseudo-random number generator to describe in code the configuration of the couplings between said encipher functional modules.

10. Encipher apparatus according to claim 9, wherein said generating means is operative to use a respective random or pseudo-random number generated by said random or pseudo-random number generator to describe in code the configuration of inputs and outputs of a respective said encipher functional module.

15

11. Encipher apparatus according to any preceding claim, wherein each said encipher functional module comprises a logic gate which does not conserve logic.

20 12. Encipher apparatus according to any preceding claim, wherein said plurality of encipher functional modules form a programmable circuit.

13. Encipher apparatus according to claim 12, wherein  
said plurality of encipher functional modules comprise  
a programmable logic gate array, and said configuring  
means comprises a programming means for programming said  
5 programmable logic gate array.

14. Encipher apparatus according to claim 12, wherein  
said encipher functional modules comprise analogue  
electronic modules.

10

15. Encipher apparatus according to any one of claims  
1 to 11, wherein said signal is an optical signal and  
said encipher functional modules comprise optical  
components.

15

16. Encipher apparatus according to any one of claims  
1 to 11, comprising a programmable computing apparatus,  
wherein said encipher functional modules comprise a  
computer code routine implemented on said programmable  
20 computing apparatus.

17. Encipher apparatus according to claim 16, wherein said encipher functional modules comprise a computer code routine repeatedly implemented dependent upon configuration information from said configuring means.

5

18. Encipher apparatus according to claim 16 or claim 17, wherein said computer code routine implements a logic gate.

10 19. Encipher apparatus according to any preceding claim including first selection means for selecting a type of encipher functional module to be used from amongst a plurality of possible types of encipher functional modules, wherein said configuring means is adapted to  
15 configure the encipher apparatus to use the selected type of encipher functional module.

20. Encipher apparatus according to any preceding claim, including second selection means for selecting the number  
20 of said encipher functional modules to be used, wherein said configuring means is adapted to configure the

encipher apparatus to use the selected number of encipher functional modules.

21. Encipher apparatus according to any preceding claim  
5 including third selection means for selecting the number of said inputs and said outputs for said encipher functional modules, wherein said configuring means is adapted to configure said encipher functional modules to have the selected number of inputs and outputs.

10

22. Encipher apparatus according to any preceding claim including splitting means for splitting the signal across the inputs of a first said encipher functional module in the sequence.

15

23. A method of enciphering a signal, the method comprising:

configuring couplings between a plurality of inputs and a plurality of outputs between a plurality of  
20 encipher functional modules coupled sequentially; and  
passing said signal through said encipher functional modules;

wherein each encipher functional module carries out a reversible operation on said signal and said encipher functional modules act sequentially on the signal.

5 24. A method according to claim 23, wherein the encipher functional modules are of a single type.

25. A method according to claim 23 or claim 24, wherein each said encipher functional module comprises a  
10 controllable switch module, and at least one of the inputs controls the switching operation.

26. A method according to any one of claims 23 to 25, wherein said encipher functional modules each act as a  
15 reversible gate.

27. A method according to any one of claims 23 to 26, wherein the couplings between said encipher functional modules are configured in accordance with information  
20 describing the configuration of the couplings between said encipher functional modules.



28. A method according to claim 27, including receiving said information.

29. A method according to claim 27, including generating  
5 said information.

30. A method according to claim 27, including generating random or pseudo-random numbers and using the generated random or pseudo-random numbers to describe in code the  
10 configuration of the couplings between said encipher functional modules.

31. A method according to claim 30, wherein a respective generated random or pseudo-random number is used to  
15 described in code the configuration of inputs and outputs of a respective said encipher functional module.

32. A method according to any one of claims 29 to 31, wherein said encipher function modules perform logic  
20 operations on said signal.

33. A method according to claim 32, wherein the logic operations to not conserve logic.

34. A method according to any one of claims 23 to 33,  
5 wherein said encipher functional modules comprise a programmable logic gate array and the configuring step includes programming said programmable logic gate array.

35. A method according to any one of claims 23 to 33,  
10 implemented by computer code on a computing apparatus, wherein said encipher functional modules comprise a computer code routine implemented dependent upon configuration information.

15 36. A method according to claim 35, wherein the computer code routine is implemented repeatedly dependent upon the number of said encipher functional units to be implemented.

20 37. A method according to any one of claims 23 to 36, including selecting the type of encipher functional

module to be used from amongst a plurality of possible types of encipher functional modules.

38. A method according to any one of claims 23 to 37,  
5 including selecting the number of said encipher functional modules used.

39. A method according to any one of claims 23 to 38,  
including selecting the number of inputs and the number  
10 of outputs for said encipher functional modules.

40. A method according to any one of claims 23 to 39,  
including splitting the signal across the inputs of a first of said encipher functional modules in the  
15 sequence.

41. Decipher apparatus for deciphering an enciphered signal, comprising:

a plurality of decipher functional modules  
20 sequentially coupled to operate sequentially on the signal, each decipher functional module having a plurality of inputs and a plurality of outputs and being

operable to carry out a reversible operation on the enciphered signal; and

configuring means for configuring couplings between the plurality of outputs and inputs between said decipher  
5 functional modules dependent upon the enciphering of said enciphered signal.

42. Decipher apparatus to claim 41, wherein said decipher functional modules are of a single type.

10

43. Decipher apparatus according to claim 41 or claim 42, wherein said configuring means is operative to use information describing the configuration of the couplings between said decipher functional modules.

15

44. Decipher apparatus according to claim 43, wherein said information describing the configuration of the couplings between said decipher functional modules is equivalent to a reverse description of information  
20 describing the configuration of the couplings between encipher functional modules used to encipher the enciphered signal.

45. Decipher apparatus according to claim 43 or claim 44, including means for receiving said information.

46. Decipher apparatus according to claim 43 or claim  
5 44, including means for generating said information.

47. Decipher apparatus according to claim 46, wherein the generating means includes a random or pseudo-random number generator and is operative to use random or  
10 pseudo-random numbers generated by said random or pseudo-random number generator to describe in code the configuration of the couplings between said decipher functional modules.

15 48. Decipher apparatus according to claim 47, wherein said generating means is operative to use a respective random or pseudo-random number generated by said random or pseudo-random number generator to describe in code the configuration of the inputs and outputs of a respective  
20 said decipher functional module.

49. Decipher apparatus according to any one of claims 41 to 48, wherein each said decipher functional module comprises a controllable switch module and at least one said input acts to control a switching operation.

5

50. Decipher apparatus according to any one of claims 41 to 49, wherein each said decipher functional modules comprises a reversible gate.

10 51. Decipher apparatus according to claim 50, wherein said reversible gate comprises a Fredkin's gate or an AND/NAND gate.

15 52. Decipher apparatus according to any one of claims 41 to 51, wherein each said decipher functional module comprises a logic gate which does not conserve logic.

20 53. Decipher apparatus according to any one of claims 41 to 52, wherein said plurality of decipher functional modules form a programmable circuit.

54. Decipher apparatus according to claim 53, wherein said plurality of decipher functional modules comprise a programmable logic gate array, and said configuring means comprises a programming means for programming said  
5 programmable logic gate array.

55. Decipher apparatus according to any one of claims 41 to 52, wherein said decipher functional modules comprise analogue electronic modules.

10

56. Decipher apparatus according to any one of claims 41 to 52, wherein said signal is an optical signal and said decipher functional modules comprise optical components.

15

57. Decipher apparatus according to any one of claims 41 to 52, comprising a programmable computing apparatus, wherein said decipher functional modules comprise a computer code routine implemented on said programmable  
20 computing apparatus.

58. Decipher apparatus according to claim 57, wherein said decipher functional modules comprise a computer code routine repeatedly implemented dependent upon configuration information from said configuring means.

5

59. Decipher apparatus according to claim 58, wherein said computer code routine implements a logic gate.

60. Decipher apparatus according to any one of claims 10 41 to 59, wherein said configuring means is adapted to configure the type of decipher functional module dependent upon the type of encipher functional modules used in the enciphering of the enciphered signal.

15 61. Deciphering apparatus according to any one of claims 41 to 60, wherein said configuring means is adapted to configure the number of decipher functional modules dependent upon the number of encipher functional modules used in the enciphering of the enciphered signal.

20

62. Decipher apparatus according to any one of claims 41 to 61, wherein said configuring means is adapted to



configure the number of inputs and outputs of said decipher functional modules dependent upon the number of inputs and outputs of the encipher functional modules used in the enciphering of the enciphered signal.

5

63. A method of deciphering an enciphered signal, the method comprising:

configuring couplings between a plurality of inputs and outputs between a plurality of decipher functional modules coupled sequentially; and

10

passing said enciphered signal through said decipher functional modules;

wherein each said decipher functional module carries out an operation on said enciphered signal which is reversible by respective said decipher functional modules and said decipher functional modules act sequentially on said enciphered signal.

15

20

64. A method according to claim 63, wherein said decipher functional modules are of a single type.

65. A method according to claim 63 or claim 64, wherein the configuration is carried out using information describing the configuration of the couplings between said decipher functional modules.

5

66. A method according to claim 65, wherein said information describing the configuration of the couplings between said decipher functional modules is equivalent to a reverse description of information describing the configuration of the couplings between encipher functional modules used to encipher the enciphered signal.

67. A method according to any one of claims 63 to 65, including receiving said information.

68. A method according to any one of claims 63 to 65, including generating said information.

69. A method according to claim 68, including generating random or pseudo-random numbers and using the generated random or pseudo-random numbers to describe in code the

configuration of the couplings between said decipher functional modules.

70. A method according to claim 69, wherein a respective  
5 generated random or pseudo-random number is used to  
described in code the configuration of the inputs and  
outputs a respective said decipher functional module.

71. A method according to any one of claims 63 to 70,  
10 wherein said decipher functional module is a controllable  
switch module, and at least one of the inputs controls  
the switching operation.

72. A method according to any one of claims 63 to 71,  
15 wherein said decipher functional modules each act as a  
reversible gate.

73. A method according to any one of claims 63 to 72,  
wherein said decipher functional modules perform logic  
20 operations on the enciphered signal.

74. A method according to claim 73, wherein the logic operations do not conserve logic.

75. A method according to any one of claims 63 to 74,  
5 wherein said decipher functional modules comprise a programmable logic gate array and the configuring step includes programming said programmable logic gate array.

76. A method according to anyone of claims 63 to 74,  
10 implemented by computer code on a computing apparatus, wherein said decipher functional modules comprise a computer code routine implemented dependent upon configuration information.

15 77. A method according to claim 76, wherein the computer code routine is implemented repeatedly dependent upon the number of said decipher functional units to be implemented.

20 78. A method according to any one of claims 63 to 77, wherein the type of decipher functional module is

configured dependent upon the type of encipher functional module used in the enciphering of the enciphered signal.

79. A method according to any one of claims 63 to 78,  
5 wherein the number of decipher modules is configured dependent upon the number of encipher modules used in the enciphering of the enciphered signal.

80. A method according to any one of claims 63 to 79,  
10 wherein the number of inputs and outputs of the decipher functional modules is configured dependent upon the number of inputs and outputs of the encipher functional modules used in the enciphering of the enciphered signal.

15 81. Apparatus for generating a cipher code, comprising:  
a random or pseudo-random number generator;  
encoding means for encoding information describing  
the configuration of couplings between cipher functional  
modules for enciphering and/or deciphering a signal using  
20 random or pseudo-random numbers generated by said  
generator; and

output means for outputting the information for use in enciphering and/or deciphering a signal.

82. Apparatus according to claim 81, including first  
5 selection means for selecting at least one type of cipher functional module to be used from amongst a plurality of possible types of cipher functional modules.

83. Apparatus according to claim 81 or claim 82,  
10 including second selection means for selecting the number of said cipher functional modules to be used, said encoding means being adapted to include the selected number in the encoded information.

15 84. Apparatus according to any one of claims 81 to 83, including third selection means for selecting the number of inputs and outputs of said cipher functional modules, said encoding means being adapted to include the selected number in the encoded information.

85. A method of generating a cipher code, the method comprising:

generating random or pseudo-random numbers;

5 encoding information describing the configuration of couplings between cipher functional modules for enciphering and/or deciphering a signal using the generated random or pseudo-random numbers; and

outputting the information for use in enciphering and/or deciphering a signal.

10

86. A method according to claim 85, including selecting a type of cipher functional module to be used from amongst a plurality of possible types of cipher functional modules.

15

87. A method according to claim 85 or claim 86, including selecting the number of said cipher functional modules used, said encoding means being adapted to include the selected number in the encoded information.

20

88. A method according to any one of claims 85 to 87, including selecting the number of inputs and outputs of

said cipher functional modules, said encoding means being adapted to include the selected number in the encoded information.

5 89. Cipher apparatus comprising the encipher apparatus of any one of claims 1 to 22 and the decipher apparatus of any one of claims 41 to 62, wherein said encipher functional modules and said decipher functional modules comprise the same functional cipher modules but implement  
10 in opposite order.

90. A cipher method for enciphering and deciphering a signal comprising the encipher method of any one of claims 23 to 40 and the decipher method of any one of  
15 claims 63 to 80.

91. Processor implementable instructions for controlling a processor to carry out the method of any one of claims 23 to 40, 63 to 80, 85 to 89 or 90.  
20

92. A carrier medium carrying the processor implementable instructions according to claim 91.



93. A storage medium storing logic to configure a programmable logic gate array to carry out the method of any one of claims 23 to 40, 63 to 80, 85 to 89 or 90.

1/17

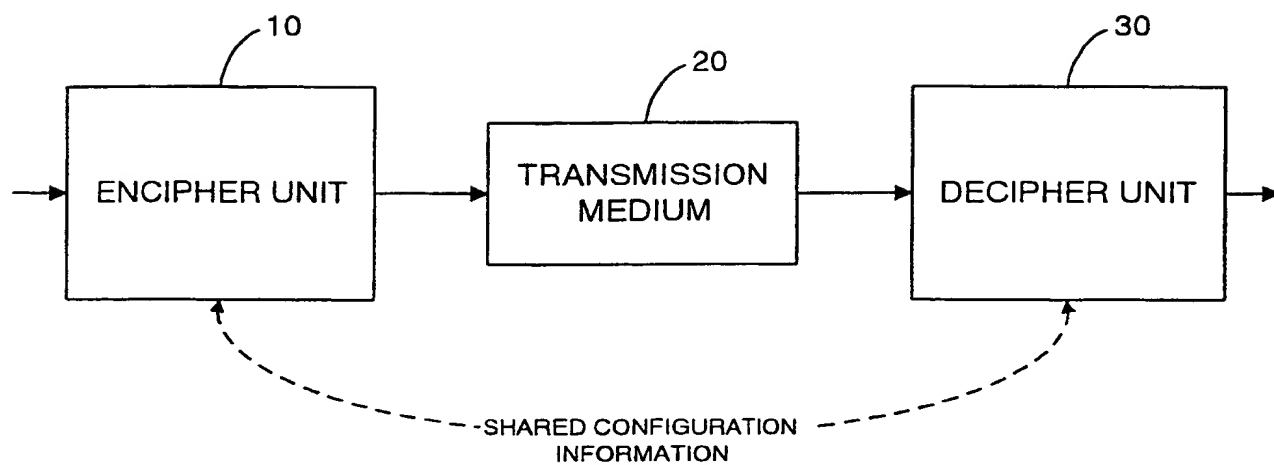


Fig 1

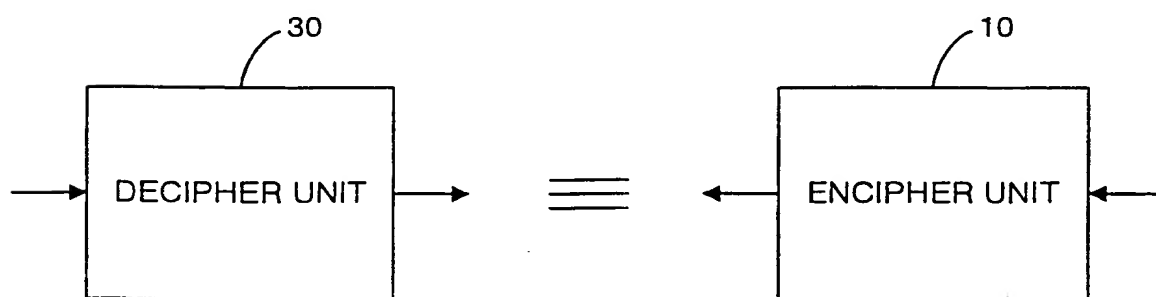


Fig 2

2/17

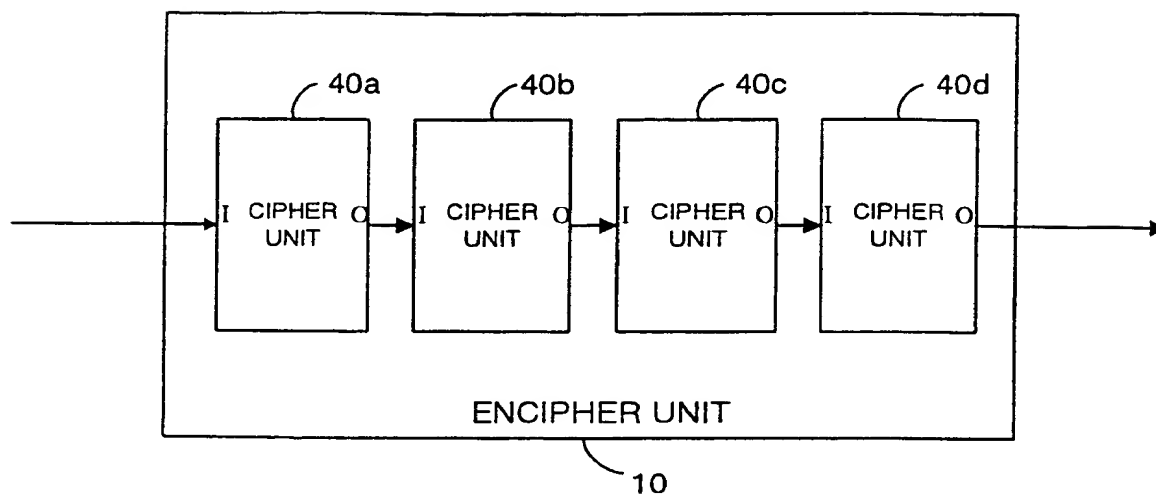


Fig 3

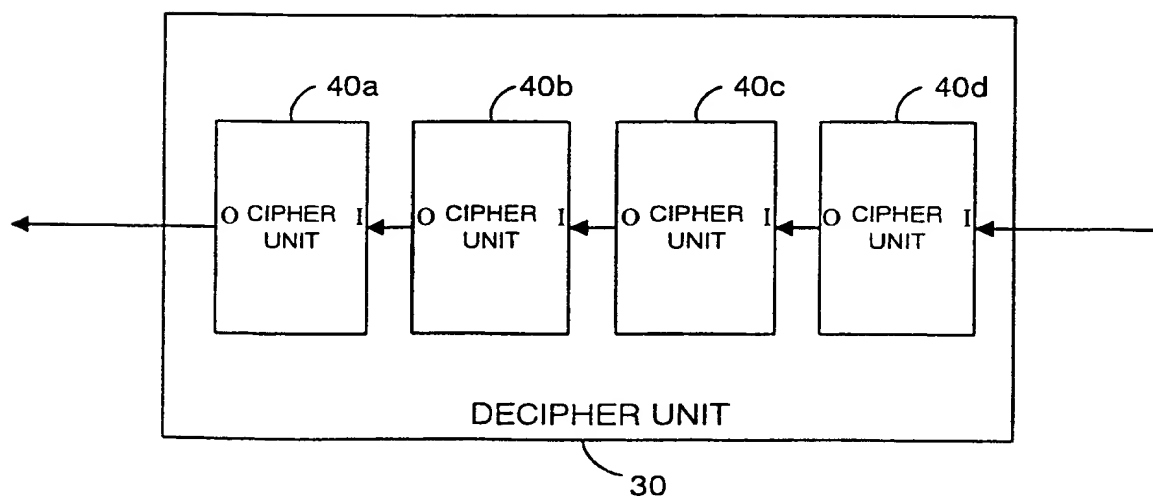


Fig 4

3/17

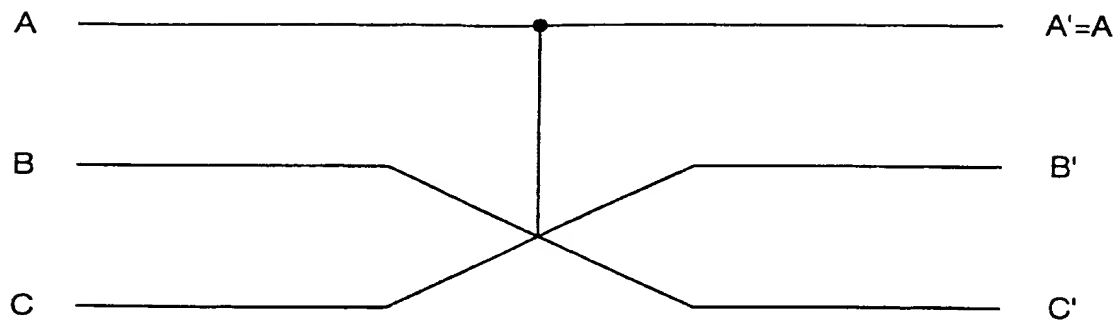


Fig 5

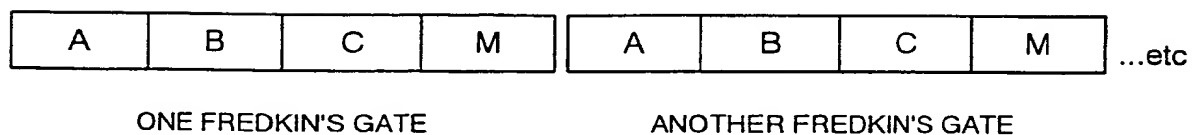


Fig 6

4/17

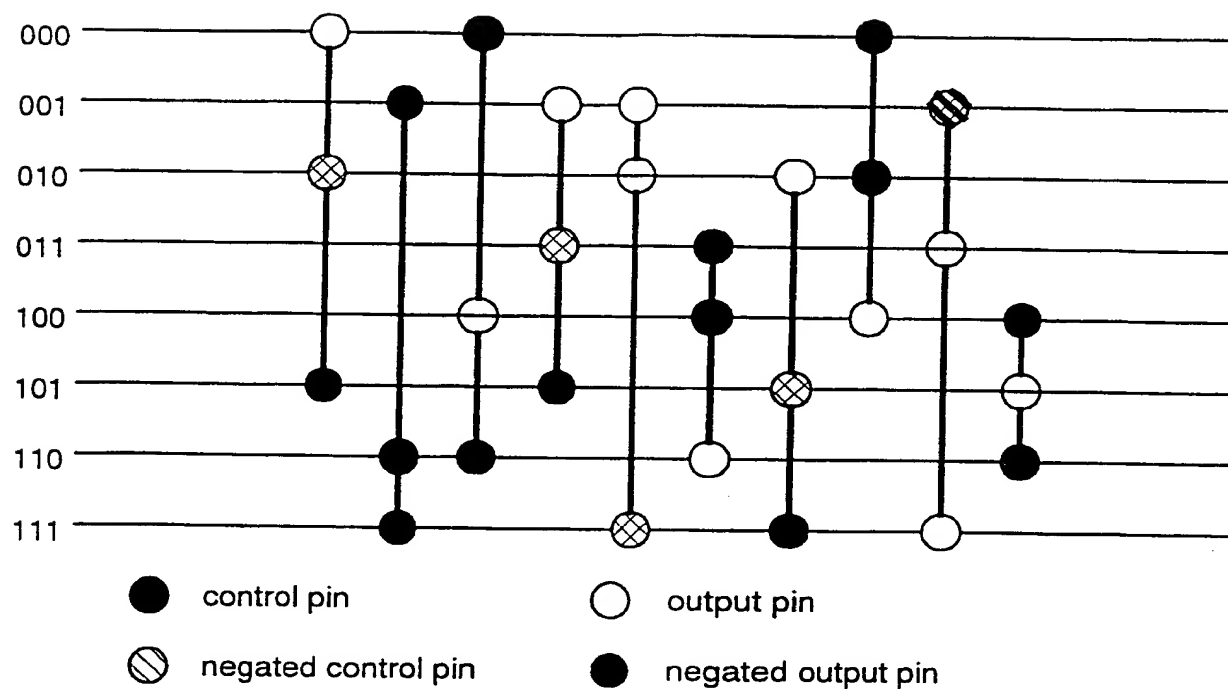


Fig 7

gate 1                      gate 2                      gate M

01000010101001    110001111011 ... 001011111000    110100101110

Fig 8

5/17

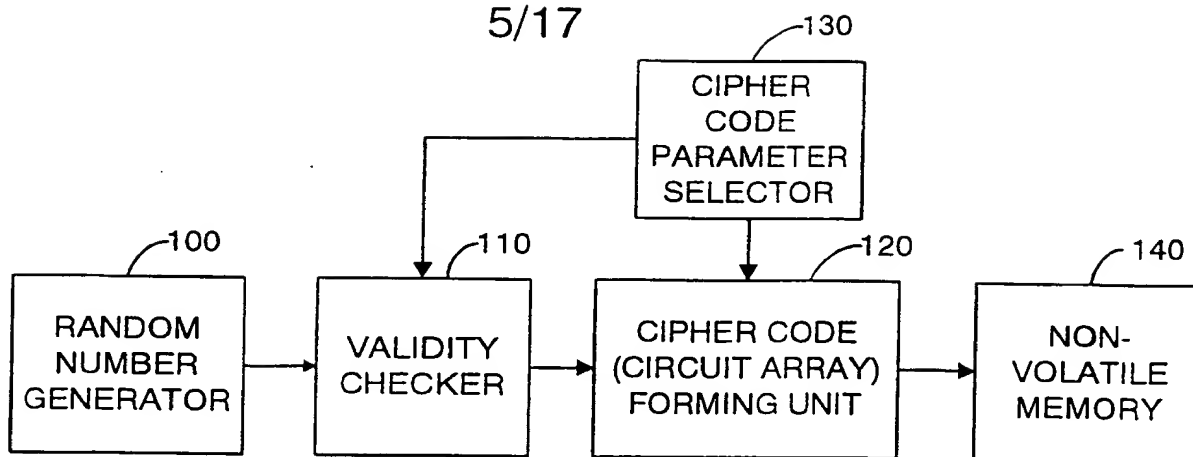


Fig 9

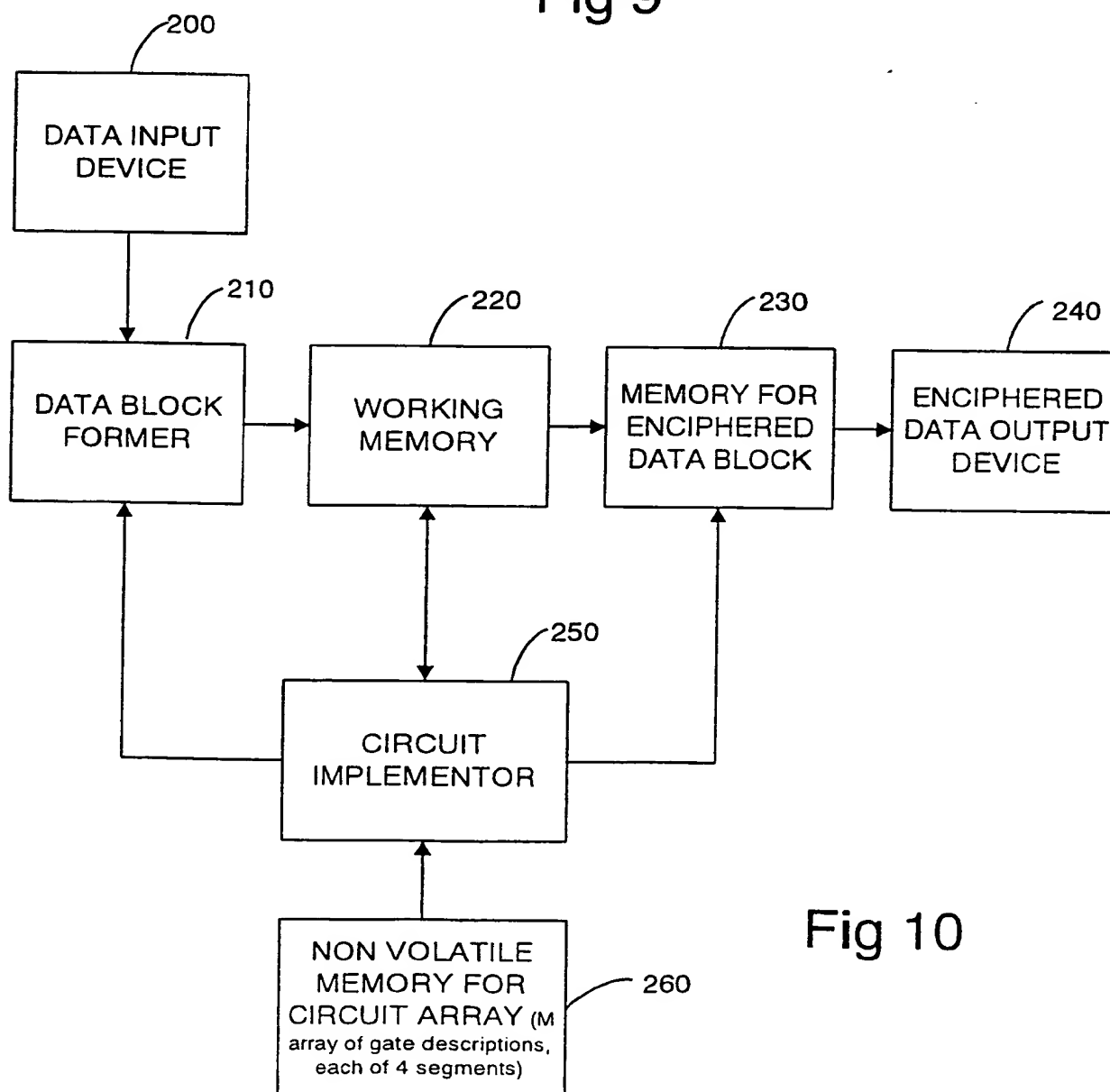


Fig 10

6/17

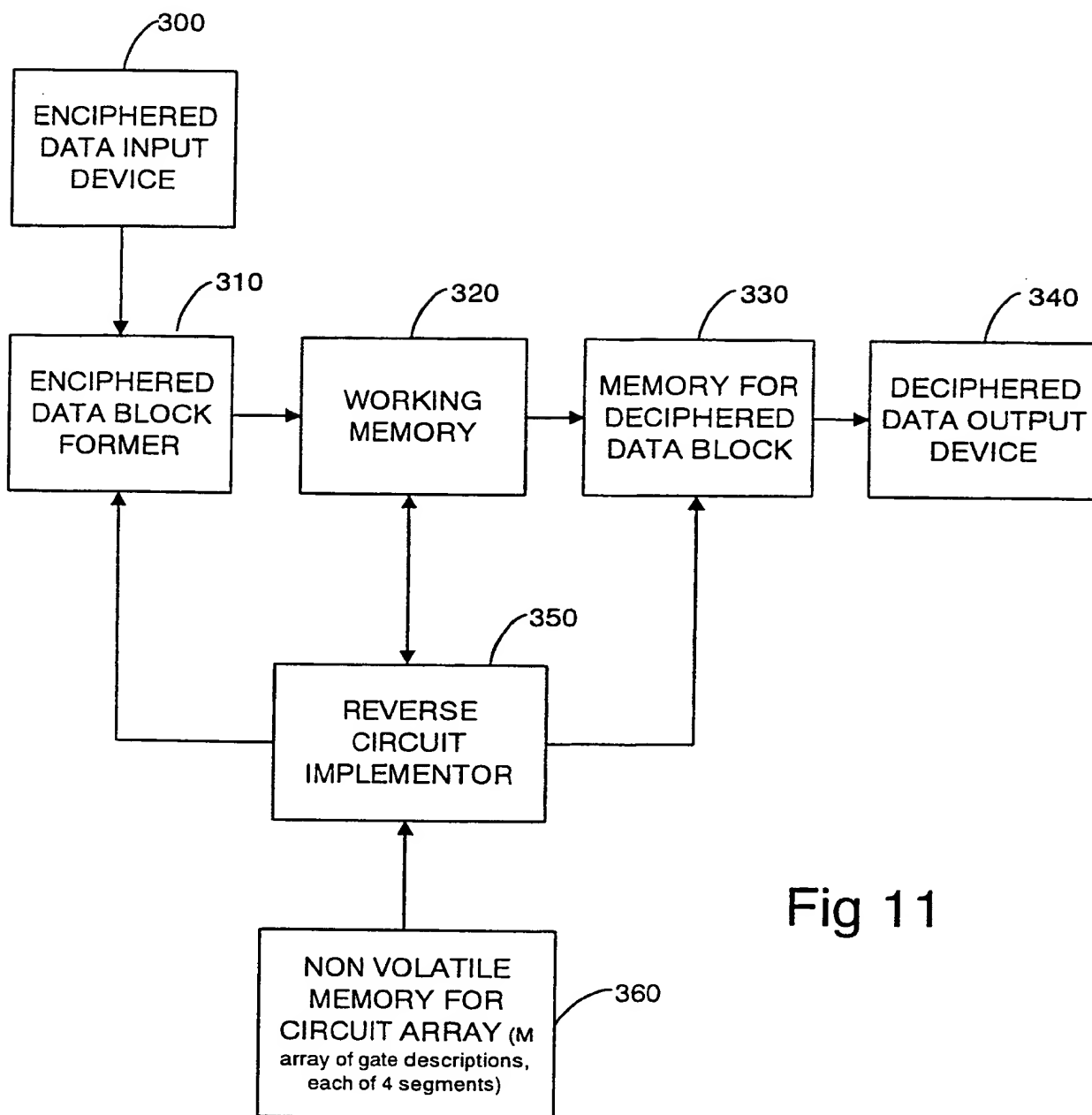


Fig 11

7/17

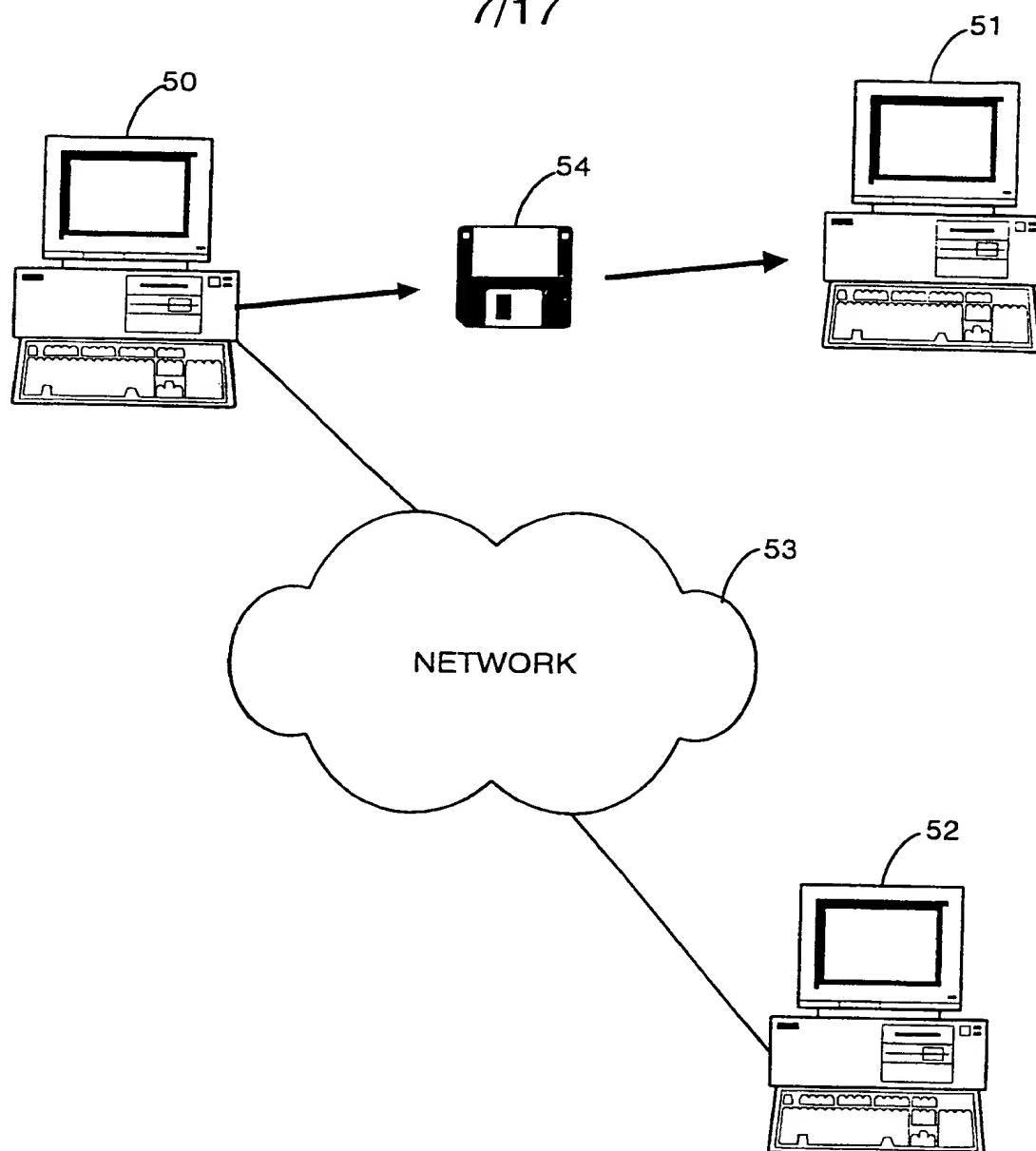


Fig 12



8/17

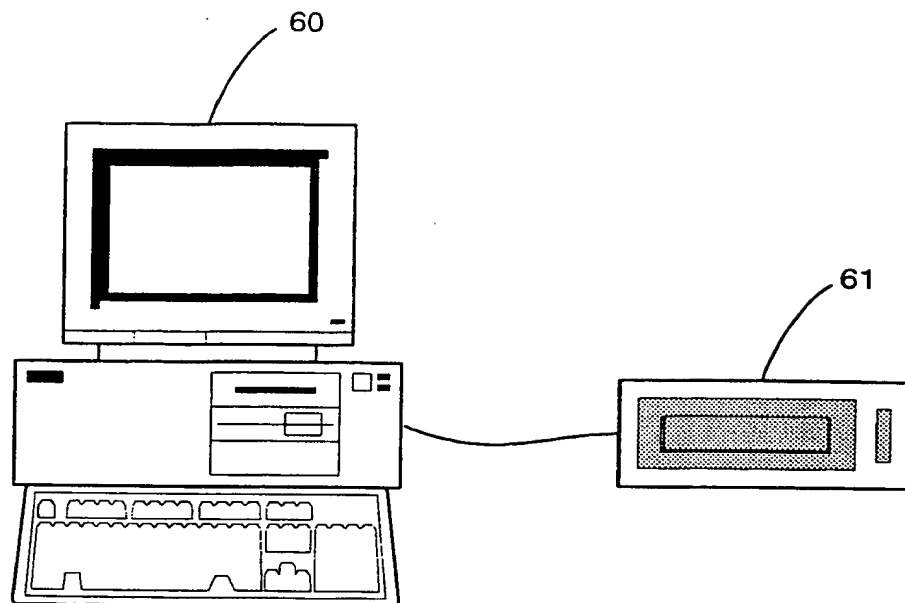


Fig 13

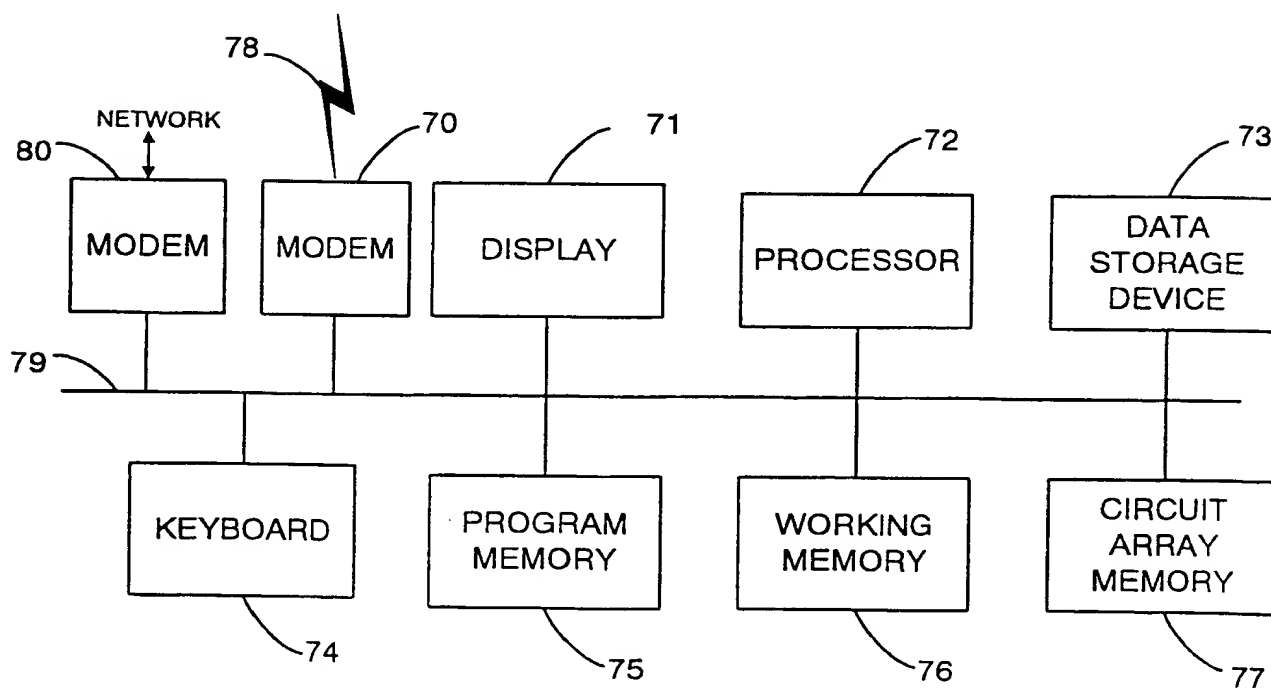
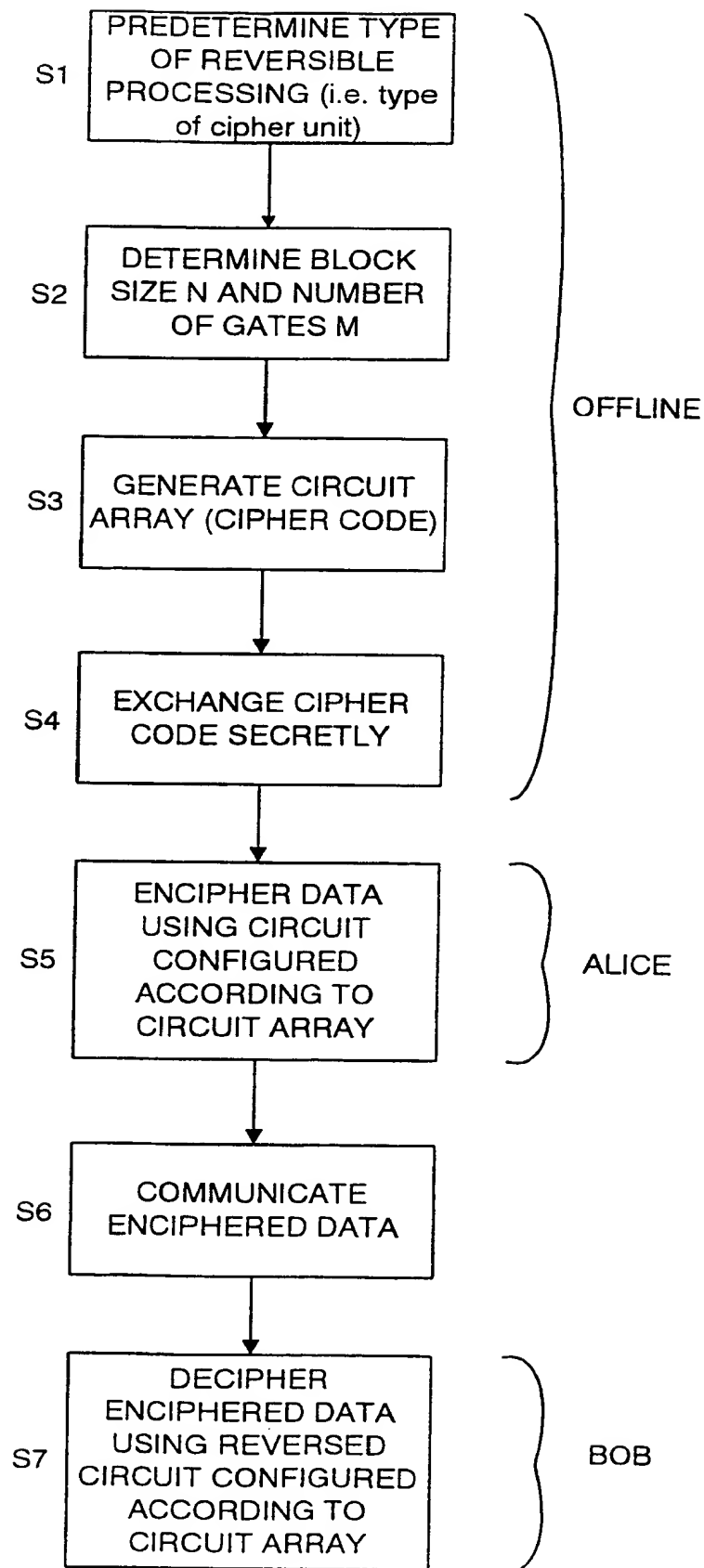


Fig 14

9/17

Fig 15



10/17

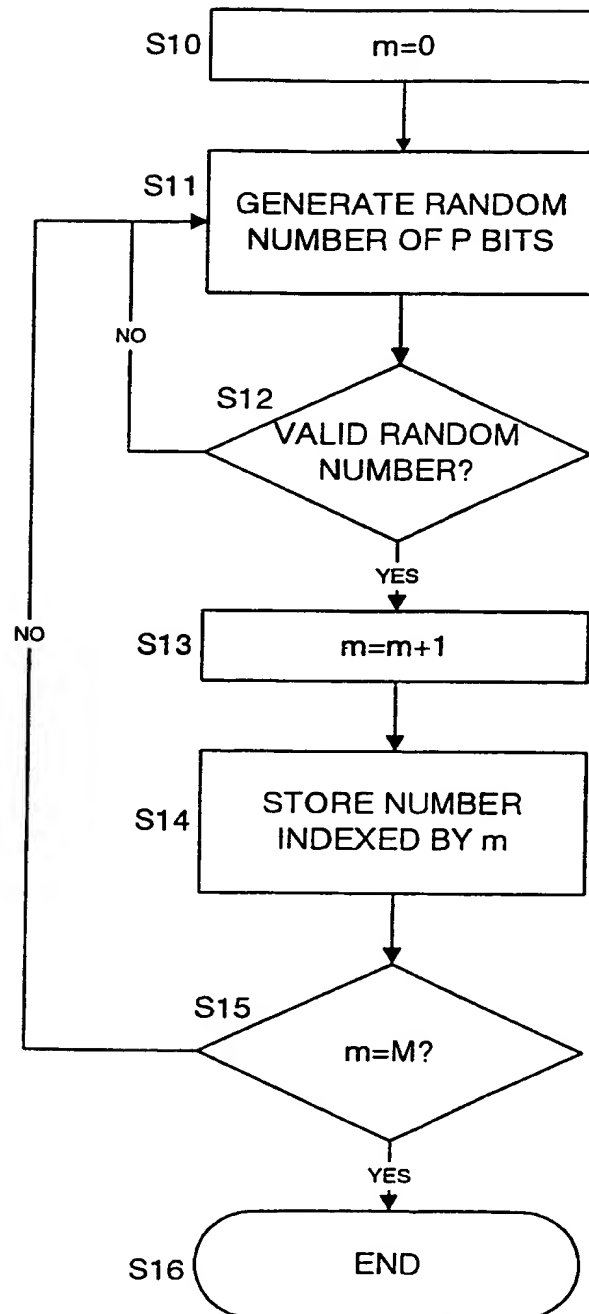


Fig 16a

11/17

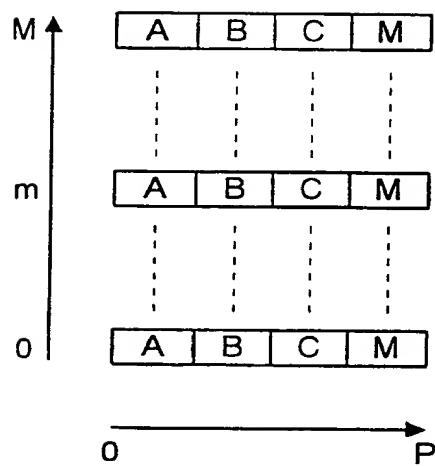
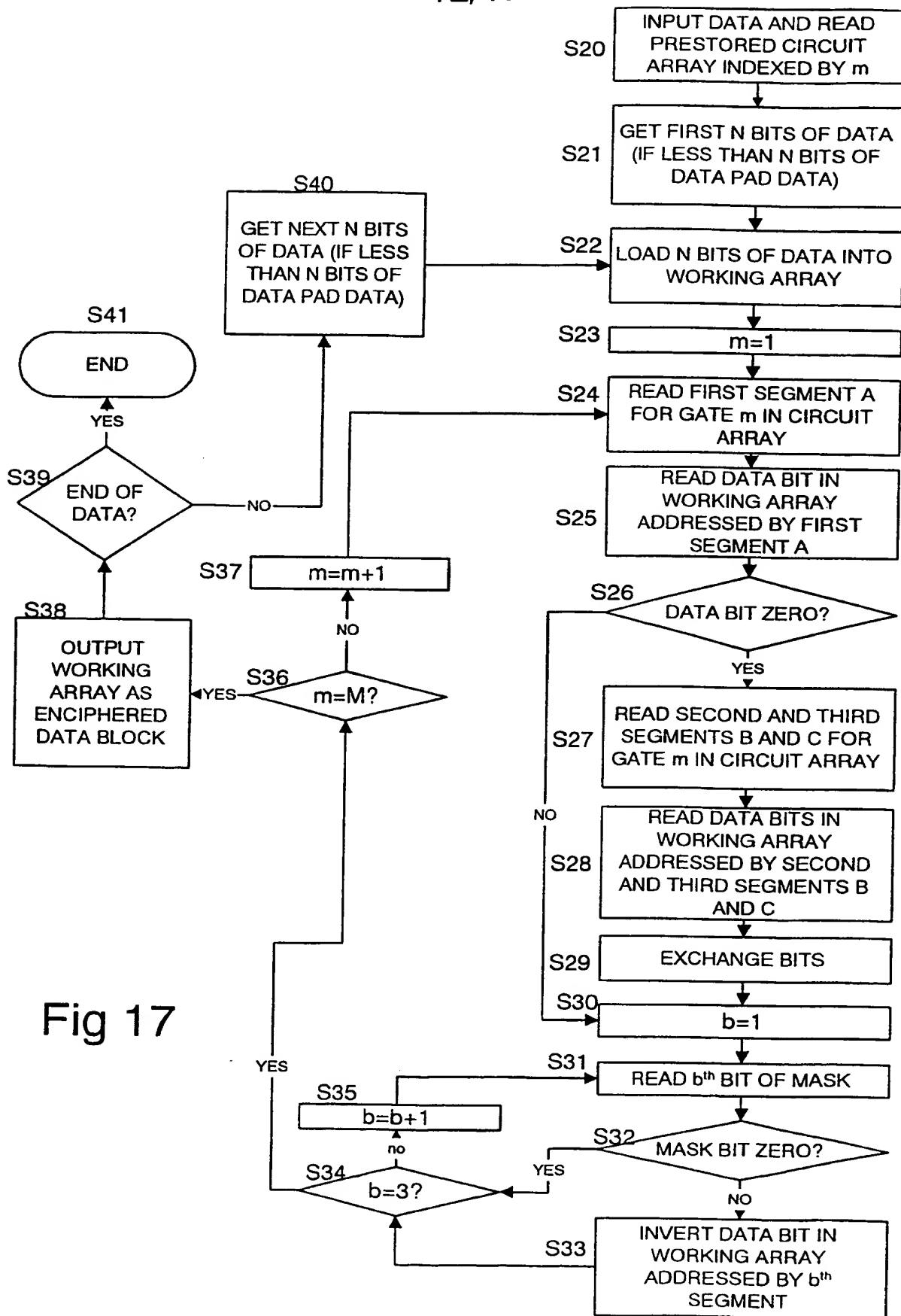


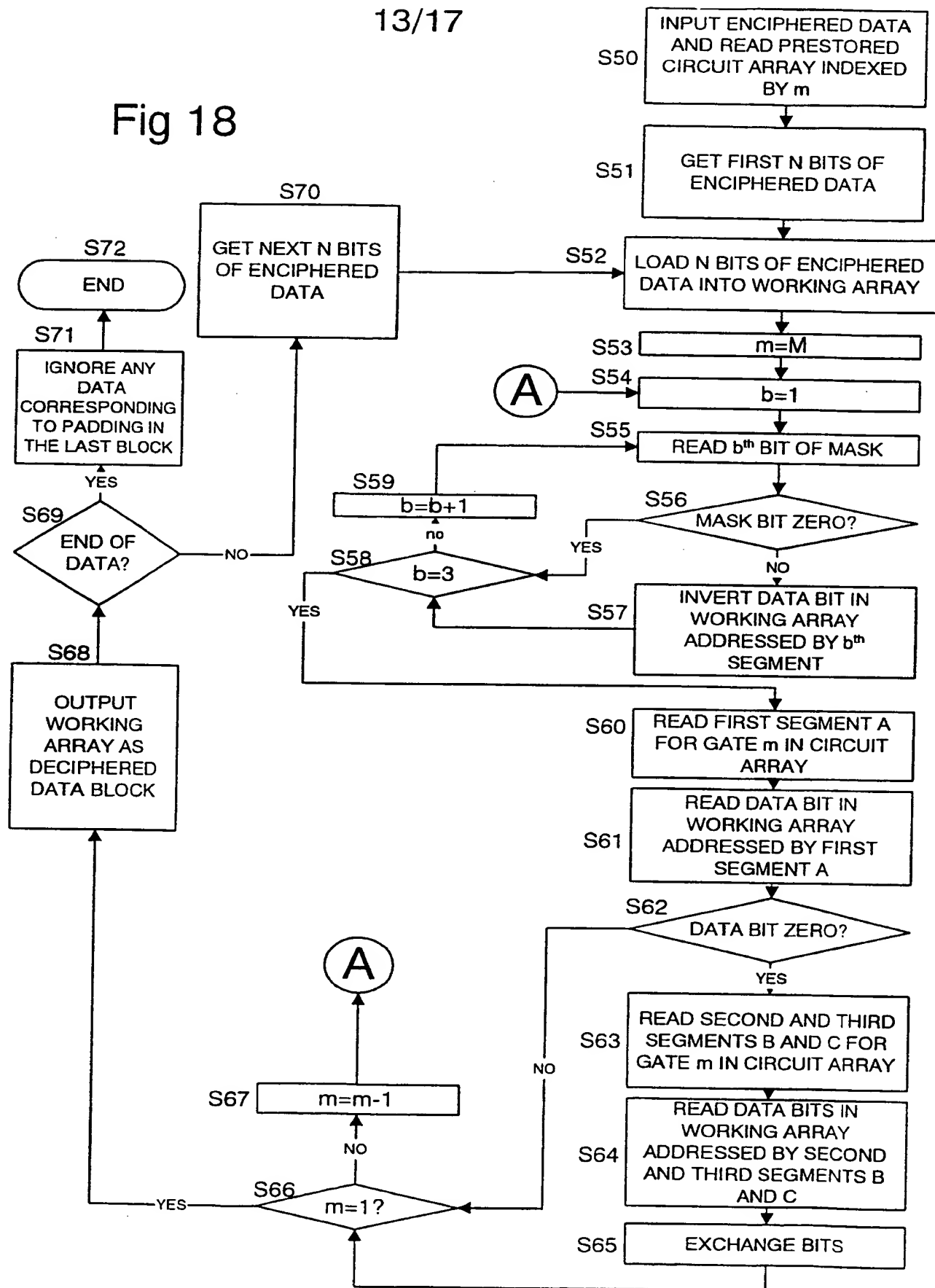
Fig 16b

12/17



13/17

Fig 18



14/17

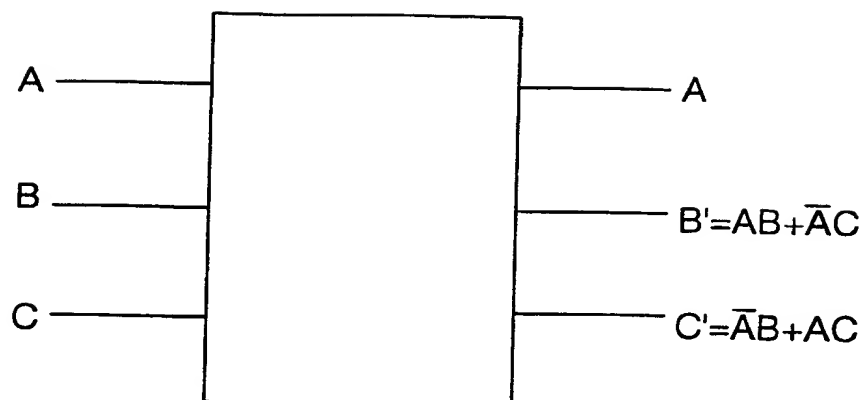


Fig 19

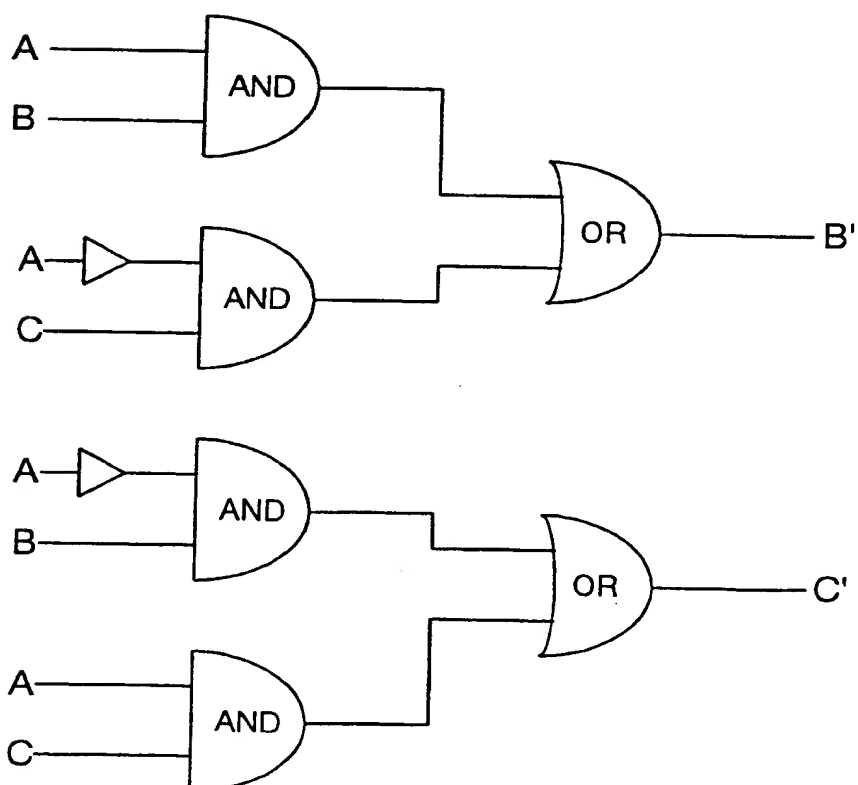


Fig 20

15/17

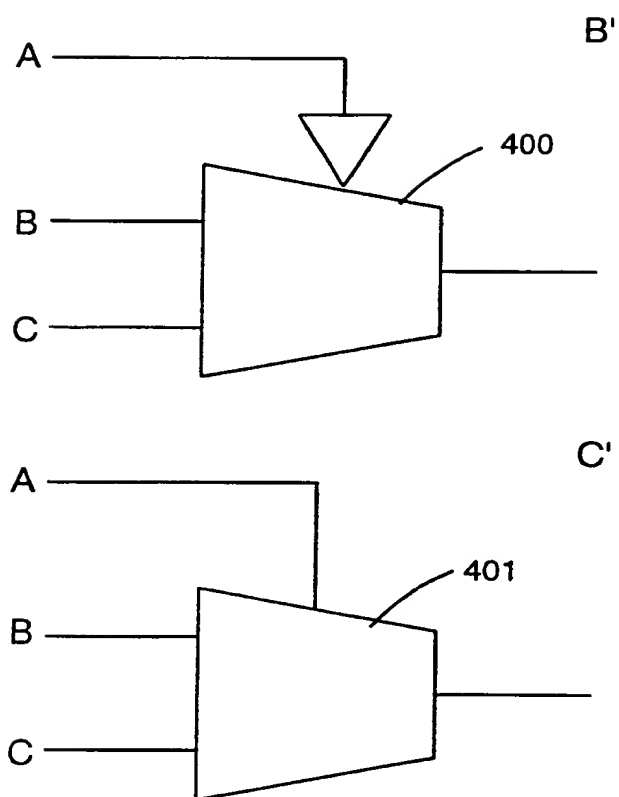


Fig 21



16/17

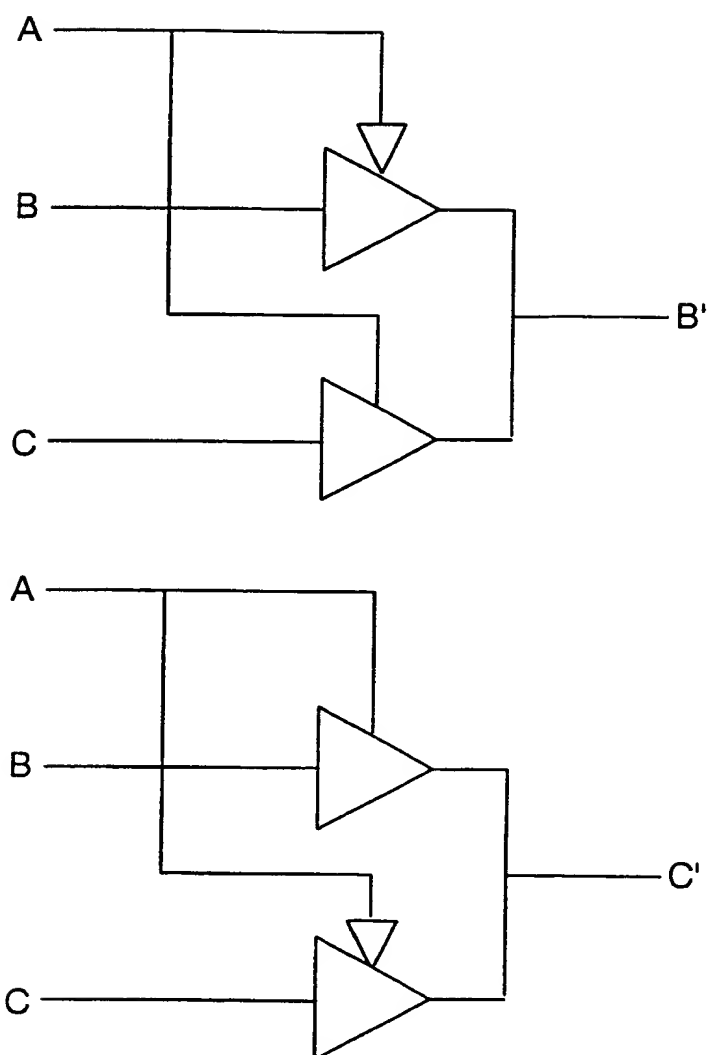


Fig 22

17/17

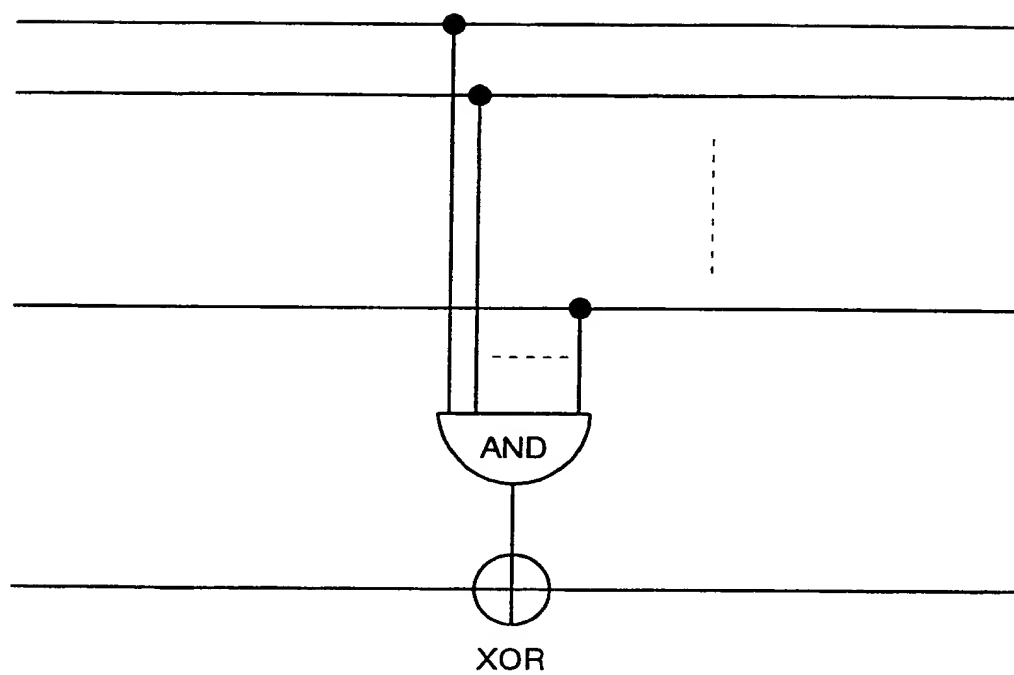


Fig 23

## INTERNATIONAL SEARCH REPORT

Inter Application No  
PCT/GB 99/03891

A. CLASSIFICATION OF SUBJECT MATTER  
IPC 7 H04L9/06 H04L9/00

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>NANDI S ET AL: "THEORY AND APPLICATIONS OF CELLULAR AUTOMATA IN CRYPTOGRAPHY" IEEE TRANSACTIONS ON COMPUTERS, NEW YORK (US), vol. 43, no. 12, 1 December 1994 (1994-12-01), pages 1346-1356, XP000484159 abstract page 1348, right-hand column, line 6 - line 34 page 1351, right-hand column, last paragraph -page 1352, left-hand column, line 16</p> <p style="text-align: center;">— -/-</p>	<p>1,3,12, 23,25, 41,49, 53,63,71</p>

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

## \* Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

20 March 2000

Date of mailing of the international search report

27/03/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Holper, G

# INTERNATIONAL SEARCH REPORT

International Application No.

PCT/GB 99/03891

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>WO 89 07375 A (MOTOROLA INC) 10 August 1989 (1989-08-10)</p> <p>page 4, line 1 -page 5, line 30 page 6, line 1 - line 18</p>	<p>1,12,13, 23,34, 41,53, 54,63,75</p>
A	<p>EP 0 267 647 A (PHILIPS) 18 May 1988 (1988-05-18) column 2, last line -column 3, line 13 column 4, line 1 - line 45 column 5, line 1 - line 28</p>	<p>1,23,41, 63</p>
A	<p>PATENT ABSTRACTS OF JAPAN vol. 017, no. 456 (E-1418), 20 August 1993 (1993-08-20) &amp; JP 05 102960 A (NEC CORP), 23 April 1993 (1993-04-23) abstract</p>	<p>7-10, 28-31, 45-48, 67-70</p>
A	<p>US 4 984 271 A (GOTO) 8 January 1991 (1991-01-08) abstract; figure 3</p>	<p>1,23,41</p>

# INTERNATIONAL SEARCH REPORT

information on patent family members

International Application No

PCT/GB 99/03891

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 8907375	A	10-08-1989	US 4914697 A	03-04-1990
			AT 139392 T	15-06-1996
			CA 1336721 A	15-08-1995
			DE 68926670 D	18-07-1996
			DE 68926670 T	19-12-1996
			EP 0398931 A	28-11-1990
			HK 1004585 A	27-11-1998
			JP 3500117 T	10-01-1991
			KR 9614682 B	19-10-1996
EP 267647	A	18-05-1988	NL 8602847 A	01-06-1988
			AU 611653 B	20-06-1991
			AU 8095087 A	12-05-1988
			CA 1291801 A	05-11-1991
			JP 2628660 B	09-07-1997
			JP 63135035 A	07-06-1988
			US 4890324 A	26-12-1989
JP 05102960	A	23-04-1993	NONE	
US 4984271	A	08-01-1991	JP 63278438 A	16-11-1988

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference 5285399		See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416) <b>FOR FURTHER ACTION</b>
International application No. PCT/GB99/03891	International filing date (day/month/year) 23/11/1999	Priority date (day/month/year) 23/11/1998
International Patent Classification (IPC) or national classification and IPC H04L9/06		
Applicant BRITISH TELECOMMUNICATIONS PUBLIC LIMITED .. et al		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.



2. This REPORT consists of a total of 5 sheets, including this cover sheet.

☒ This report is also accompanied by ANNEXES, i.e. sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

These annexes consist of a total of 31 sheets.

3. This report contains indications relating to the following items:

- I ☒ Basis of the report
- II ☐ Priority
- III ☐ Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- IV ☐ Lack of unity of invention
- V ☒ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
- VI ☐ Certain documents cited
- VII ☒ Certain defects in the international application
- VIII ☐ Certain observations on the international application

Date of submission of the demand 09/05/2000	Date of completion of this report
Name and mailing address of the international preliminary examining authority:  European Patent Office D-80298 Munich Tel. +49 89 2399 - 0 Tx: 523656 epmu d Fax: +49 89 2399 - 4465	Authorized officer Cretaine, P Telephone No. +49 89 2399 8828 

# INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No. PCT/GB99/03891

## I. Basis of the report

1. This report has been drawn on the basis of *(substitute sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to the report since they do not contain amendments (Rules 70.16 and 70.17).):*

### Description, pages:

1-18 as received on 03/11/2000 with letter of 01/11/2000

### Claims, No.:

1-83 as received on 03/11/2000 with letter of 01/11/2000

### Drawings, sheets:

1/17-17/17 as originally filed

2. With regard to the **language**, all the elements marked above were available or furnished to this Authority in the language in which the international application was filed, unless otherwise indicated under this item.

These elements were available or furnished to this Authority in the following language: , which is:

- ☐ the language of a translation furnished for the purposes of the international search (under Rule 23.1(b)).
- ☐ the language of publication of the international application (under Rule 48.3(b)).
- ☐ the language of a translation furnished for the purposes of international preliminary examination (under Rule 55.2 and/or 55.3).

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international preliminary examination was carried out on the basis of the sequence listing:

- ☐ contained in the international application in written form.
- ☐ filed together with the international application in computer readable form.
- ☐ furnished subsequently to this Authority in written form.
- ☐ furnished subsequently to this Authority in computer readable form.
- ☐ The statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.
- ☐ The statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished.

4. The amendments have resulted in the cancellation of:

- ☒ the description, pages: 19-25
- ☒ the claims, Nos.: 84-93

# INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No. PCT/GB99/03891

☐ the drawings, sheets:

5. ☐ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed (Rule 70.2(c)):

*(Any replacement sheet containing such amendments must be referred to under item 1 and annexed to this report.)*

6. Additional observations, if necessary:

## V. Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

### 1. Statement

Novelty (N)	Yes:	Claims	1-83
	No:	Claims	
Inventive step (IS)	Yes:	Claims	1-83
	No:	Claims	
Industrial applicability (IA)	Yes:	Claims	1-83
	No:	Claims	

2. Citations and explanations  
**see separate sheet**

## VII. Certain defects in the international application

The following defects in the form or contents of the international application have been noted:  
**see separate sheet**



**Re Item V**

**Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement**

The invention relates to an enciphering apparatus (claim 1) and method (claim 20), a deciphering apparatus (claim 36) and method (claim 55), and to an apparatus (claim 71) and method (claim 75) for generating a cipher design description.

**Prior art:**

WO-A-89 07375 discloses a cryptographic method and apparatus which permit the cipher algorithm to be changed after the manufacture of the apparatus by programming electronically erasable, programmable array logic devices to perform a specific Boolean algebra function. This allows the encryption hardware to be manufactured without strict security control. However the encryption process used is fixed although the algorithm is selectable, i.e the process is not scalable. Furthermore the only security is provided by the key which is secretly exchanged between the two parties.

**Invention:**

The aim of the invention is to provide an enciphering method wherein the secret cipher code which is required for both enciphering and deciphering is information which describes the process used to carry out the enciphering.

This is achieved, according to the features of claim 1, by forming the signal to be enciphered as a sequence of data blocks and sequentially treating the blocks in a plurality of cascaded encipher functional modules. Each of the module is configurable to allow it to operate a reversible process on predetermined bit positions of a received data block, the description of these predetermined bit positions for each module representing a cipher design description which may be exchanged secretly between a sender and a receiver as a secret cipher code.

Said solution is not disclosed or suggested by the documents cited in the International Search Report. Therefore claim 1 meets the requirements of Article 33 PCT.

**INTERNATIONAL PRELIMINARY  
EXAMINATION REPORT - SEPARATE SHEET**

---

International application No. PCT/GB99/03891

Independent claim 20 relates to the method corresponding to claim 1.  
Independent claim 36 relates to a decipher apparatus performing the reverse operation of the enciphering apparatus of claim 1.  
Independent claim 55 relates to the method corresponding to claim 36.  
Independent claim 71 relates to an apparatus for randomly generating the above-mentioned cipher design description.  
Independent claim 75 relates to the method corresponding to claim 71.  
Claims 81 and 83 are program claims corresponding to method claims 20-35 or 55-70 or 75-78 or 80.  
Claim 82 relates to a medium with said program on it.

The features of each of these claims are neither disclosed nor suggested by the documents cited in the International Search Report. Therefore claims 20, 36 55, 71, 75 81-83 meets the requirements of Article 33 PCT.

Claims 2-19, 21-35, 37-54, 56-70, 72-74, 76-80 are dependent claims and as such also meet the requirements of the PCT with respect to novelty and inventive step.

**Re Item VII**

**Certain defects in the international application**

Contrary to the requirements of Rule 5.1(a)(ii) PCT, the relevant background art disclosed in the document D1 is not mentioned in the description, nor is this document identified therein.

The features of the claims are not provided with reference signs placed in parentheses (Rule 6.2(b) PCT).

## A CIPHER

The present invention generally relates to a cipher and in particular to a cipher in which the secret cipher code which is required for both enciphering and deciphering is information which describes the process used to carry out enciphering.

Two commonly used types of cryptographic algorithms are private key algorithms which use a single shared key and public key algorithms which use two keys: a public key and a private key.

In these prior art algorithms the encryption process used is fixed although the particular encryption process can be selectable by user e.g. by using a particular encryption program (algorithm). The security of the encryption is provided by the key which is secretly exchanged between the encrypter operator and decrypter operator. Such currently implemented ciphers are not easily scalable since they are defined for a specific block size and key size. In many instances the key is not big enough e.g. many ciphers have only 64 bit keys.

In accordance with a first aspect, the present invention provides encipher apparatus for enciphering a signal, comprising:

forming means for receiving the signal to be enciphered and for outputting the signal as a sequence of data blocks, each having a first predetermined number of bits;

a plurality of encipher functional modules sequentially coupled to operate sequentially on the sequence of data blocks from the forming means; and

configuring means,

wherein each encipher functional module comprises

a module input,

a module output, and

a respective data processing unit having a parallel input and a corresponding parallel output and being arranged to perform a respective reversible process upon a set of bits at its parallel input and to produce at its corresponding parallel output a corresponding enciphered set of bits,

and is operable under the control of the configuring means to couple a respective predetermined set of the bits of a data block received at its module input to the parallel input of its data processing unit and to provide at its module output an

enciphered data block in which said respective predetermined set of bits is replaced by the corresponding enciphered set of bits produced at the parallel output of its data processing unit.

Another aspect of the present invention provides a method of enciphering a  
5 signal, the method comprising:

receiving the signal to be enciphered and forming the signal into a sequence of data blocks, each having a first predetermined number of bits;

applying the sequence of data blocks to a plurality of encipher functional modules sequentially coupled to operate sequentially on the sequence of data blocks,  
10 each encipher functional module comprising

a module input,

a module output, and

a respective data processing unit having a parallel input and a corresponding parallel output and being arranged to perform a respective reversible process upon a set of  
15 bits at its parallel input and to produce at its corresponding parallel output a corresponding enciphered set of bits; and

configuring each encipher functional module to couple a respective predetermined set of the bits of a data block received at its module input to the parallel input of its data processing unit and to provide at its module output an  
20 enciphered data block in which said respective predetermined set of bits is replaced by the corresponding enciphered set of bits produced at the parallel output of its data processing unit.

Thus the present invention provides a universal cipher which is capable of implementing any cipher process. The encryption which is carried out on the signal  
25 is dependent upon the respective predetermined sets of bits. The respective predetermined sets of bits are configurable without changing the data processing unit of the cipher units. This configuration is freely selectable and is preferably selected randomly or pseudo-randomly and automatically for the usual security reasons to prevent any element of predictability.

30 In one embodiment the encipher functional modules, also referred to generically as cipher units, are identical and thus perform identical reversible operations. The invention does however encompass the use of a plurality of types of cipher units

wherein the sequential pattern of the different types is information that must be shared to allow deciphering of the signal enciphered using the pattern.

The cipher units can be implemented in many different ways such as a reversible circuit either implemented in logic gates or in logic steps performed by a computer, analog circuitry, or optical elements. In fact, the cipher units can be implemented by  
5 any physical process which is reversible.

A further aspect of the present invention provides decipher apparatus for deciphering a signal, comprising:

forming means for receiving the signal to be deciphered and for outputting  
10 the signal as a sequence of data blocks, each having a first predetermined number of bits;

a plurality of decipher functional modules sequentially coupled to operate sequentially on the sequence of data blocks from the forming means; and  
configuring means,

15 wherein each decipher functional module comprises

a module input,

a module output, and

a respective data processing unit having a parallel input and a corresponding parallel output and being arranged to perform a respective reversible process upon a set of  
20 bits at its parallel input and to produce at its corresponding parallel output a corresponding enciphered set of bits,

and is operable under the control of the configuring means to couple a respective predetermined set of the bits of a data block received at its module input to the parallel input of its data processing unit and to provide at its module output a  
25 deciphered data block in which said respective predetermined set of bits is replaced by the corresponding deciphered set of bits produced at the parallel output of its data processing unit.

Another aspect of the present invention provides a method of deciphering an enciphered signal, the method comprising:

30 receiving the signal to be deciphered and outputting the signal as a sequence of data blocks, each having a first predetermined number of bits;

applying the sequence of data blocks to a plurality of decipher functional modules sequentially coupled to operate sequentially on the sequence of data blocks, each decipher functional module comprising

- a module input,
- 5 a module output, and
- a respective data processing unit having a parallel input and a corresponding parallel output and being arranged to perform a respective reversible process upon a set of bits at its parallel input and to produce at its corresponding parallel output a corresponding enciphered set of bits; and
- 10 configuring each decipher functional module to couple a respective predetermined set of the bits of a data block received at its module input to the parallel input of its data processing unit and to provide at its module output an deciphered data block in which said respective predetermined set of bits is replaced by the corresponding deciphered set of bits produced at the parallel output of its data
- 15 processing unit.

In one embodiment the cipher units are identical and carry out identical operations on the enciphered signal.

- The use of reversible cipher units in both the encipher and decipher enable the configuration of the units to be the same although the implementation will be
- 20 reversed. It is this reversibility which allows the use of information describing the cipher process, referred to as cipher design description, to be used secretly between the encipher and decipher. In other words, instead of a secret key to be shared by the sender and receiver of an encrypted message, a cipher design description which describes the cipher process is shared instead.

- 25 Thus the invention is similar to the conventional symmetric cryptography technique except that there is no single shared key but instead a single shared cipher design description containing information describing the cipher process.

- In a similar manner to use of a private key, the cipher design description can be determined by either party in a two-party communication of an encrypted signal. In
- 30 other words, either the recipient of an encrypted signal can request for the cipher design description to be used and secretly pass this to the party for use in transmitting the encrypted signal, or the party transmitting the encrypted signal can

secretly inform the recipient of the cipher design description to be used to decrypt the signal.

The invention is equally applicable to the encryption of a signal which is not transmitted and which is instead stored securely e.g. the encryption and storage of  
5 data in a computer to prevent unauthorised access. In this example there need only be one party.

Because the cipher units are reversible, the encipher apparatus and decipher apparatus can be constituted by a single apparatus. Thus, for duplex communication of an encrypted signal or for the storage of encrypted data for retrieval and  
10 decryption it is possible for the same cipher units to be used but in reverse order for deciphering.

In one embodiment the reversible process of at least one of said data processing units is a switching operation controlled by at least one of the set of bits received at its parallel input.

15 Such a data processing unit can be implemented as a reversible gate such as a Fredkin's gate or an AND/NAND gate. Such gates can be implemented in logic either as logic gates such as a programmable circuit, in particular a programmable logic gate array, or as logic steps implementing the gates in a computer program.

An advantage of this invention is that it is inherently scalable since the number of  
20 cipher units can be varied dependent upon the configuration. Further, the size of the data block can be varied. This will also depend upon the configuration of the cipher unit.

Conveniently, ciphers are usually implemented digitally as a computer program. The programmability of a general purpose computer provides the facility for a universal  
25 cipher. Since a computer is capable of implementing a reversible computational process which implements a one-to-one mapping, and since general purpose computers are available which can be programmed to carry out any reversible computational process, any reversible computational process can be implemented thus implementing any one-to-one mapping. Any general purpose reversible computer  
30 can be used as a universal cipher. The use of a computer program to implement the reversible process further enables a user to select the type of reversible process to be implemented.

Since apparatus of the present invention can be implemented on a general purpose computer by a suitable program, the present invention can be embodied as a storage medium storing instructions, for controlling a processor e.g. a floppy disc, CD-ROM, smartcard, and programmable memory. Further, since the computer program can be transmitted over a network to be received and implemented on a computer, the present invention can be embodied as a signal carrying the processor implementable instructions. Apparatus of the present invention can also be embodied as a storage medium storing logic to configure a programmable logic gate array to carry out an encipher method or a decipher method of the present invention.

10

Embodiments of the present invention will now be described with reference to the accompanying drawings, in which:

Figure 1 is a schematic illustration of a cipher system;

Figure 2 is a schematic illustration of the reversibility of the cipher;

15 Figure 3 is a schematic illustration of an encipher apparatus of the cipher system of Figure 1;

Figure 4 is a schematic illustration of a decipher apparatus of the cipher system of Figure 1;

Figure 5 is a schematic illustration of a Fredkin's gate;

20 Figure 6 is a schematic illustration of a cipher unit design description for the configuration of a Fredkin's gate forming the basis of a cipher unit of the apparatus of Figures 3 and 4;

Figure 7 is a schematic illustration of an encipher apparatus using Fredkin's gates;

Figure 8 is an illustration of the cipher design description for the encipher apparatus of Figure 7;

25 Figure 9 is a functional diagram of a cipher design description generator;

Figure 10 is a functional diagram of an encipher apparatus;

Figure 11 is a functional diagram of a decipher apparatus;

Figure 12 is a diagram of use of the cipher system in the transmission of encrypted data;

30 Figure 13 is a diagram of a specific embodiment wherein the cipher design description and possibly the encipher/decipher apparatus is exchanged using a smartcard;



Figure 14 is a diagram of a processing apparatus capable of implementing the cipher apparatus;

Figure 15 is a flow diagram showing the generation and exchange of encrypted data using the cipher system;

- 5 Figure 16a is a flow diagram illustrating the generation of the cipher design description;

Figure 16b is a schematic diagram of the cipher design description;

Figure 17 is a flow diagram illustrating the encipher process;

Figure 18 is a flow diagram illustrating the decipher process;

- 10 Figure 19 is a Fredkin's gate illustrated as a three-input logic gate;

Figure 20 is an implementation of the logic gate of Figure 19 using AND, OR and NOT gates;

Figure 21 is a diagram of an implementation of the Fredkin's gate using multiplexers;

- 15 Figure 22 is a diagram of an implementation of the Fredkin's gate using three-state buses; and

Figure 23 is a diagram of an AND/NAND gate as an alternative reversible gate to the Fredkin's gate.

- Referring now to the drawings, Figure 1 illustrates a cipher system in general wherein an encipher unit 10; hereinafter referred to as an encipher apparatus, generates an  
20 enciphered or encrypted signal using shared configuration information, referred to as the cipher design description. The cipher design description is used to configure the encipher apparatus 10. The encrypted signal is then transmitted by a transmission medium 20 to a recipient decipher unit 30, hereinafter referred to as a decipher apparatus, which also has the shared cipher design description. The decipher  
25 apparatus 30 is configured in accordance with the cipher design description and operates the reverse of the process carried out by the encipher apparatus 10 to thereby decipher the signal.

- Although in this embodiment a transmission medium 20 is illustrated, the transmission medium could simply comprise a storage medium on which the  
30 encrypted data is stored. Thus the operator generating the encrypted signal and the operator receiving the encrypted signal may in fact be the same.

Figure 2 schematically illustrates the reversibility of a cipher apparatus to act either as a decipher apparatus 30 or an encipher apparatus 10, and the term cipher apparatus is used herein as meaning an apparatus capable of acting in either mode.

Figure 3 illustrates in more detail the encipher apparatus 10 which is comprised of 5 cipher units 40a to 40d. Although in this embodiment four cipher units are illustrated, in a practical embodiment this would typically be at least four times the block size e.g. for a block size of 128 bits the number of cipher units is at least 512. The number will however depend on the level of security desired. As can be seen in Figure 3 the input signal is received at the input I of the cipher unit 40a, which 10 provides an output signal to the input I of the cipher unit 40b, and so on, i.e. the cipher units 40a to 40d are coupled sequentially.

Figure 4 illustrates a decipher apparatus 30 in more detail. The decipher apparatus 30 comprises the same set of cipher units 40a to 40d as in the encipher apparatus 10, but they are connected in reverse sequential order, i.e. from 40d to 40a. Thus, in 15 order to decipher the enciphered signal, it is passed in sequence through the cipher units 40d to 40a, i.e. the signal to be deciphered is received at the input I of the cipher unit 40d, which provides an output signal to the input I of the cipher unit 40c, and so on.

A specific implementation of the cipher system will now be described in which the 20 cipher unit is implemented using a data processing unit or circuit known as a Fredkin's gate. Such a gate is illustrated in Figure 5. It is well known that a Fredkin's gate is both reversible (i.e. a cipher unit implemented by a Fredkin's gate can be run backwards to uncompute) and universal (i.e. can be used to design a cipher unit that implements all one-to-one mappings).

25 In the Fredkin's gate as illustrated in Figure 5, the input A is used to control the exchange of data on inputs B and C. Thus the Fredkin's gate performs a controlled exchange operation. If  $A=1$ , B and C are not exchanged i.e.  $B'=B$  and  $C'=C$ . If however,  $A=0$ ,  $B'=C$  and  $C'=B$ . In mathematical notation  $B'=AB+\bar{A}C$  and  $C'=\bar{A}B+AC$ .

30 The Fredkin's gate is a conservative logic gate i.e. it preserves the numbers of 0's and 1's from the input to the output. In a cipher system this is undesirable, thus in order to break the conservation, NOT gates are selectively applied to the outputs to

invert them. The selective inversion are operations which are inherently reversible and thus do not affect the reversibility of the circuit.

Having selected the type of reversible circuit used as the cipher unit, it is then necessary to determine a cipher unit design description to describe the arrangement  
5 of the circuits. Figure 6 illustrates one such cipher unit design description wherein each Fredkin's gate is described by a respective four segment cipher unit design description. Each of the first three segments describe input pin numbers to which the three inputs A, B and C of a gate are coupled, and, correspondingly, the output pin numbers to which the three outputs A', B' and C' of that gate are coupled. The  
10 last segment, referred to as the mask M, is used to encode a description of the presence or the absence of inverters on each of the three outputs A', B' and C'.

Consider an encipher process in which it is decided that the input signal is to be enciphered in 8 bit blocks. The input data thus comprises a 8 bit array indexed from 000 (first bit) to 111 (eighth bit). Each segment of the cipher unit design description  
15 for the gate of an encipher unit thus comprises a 3 bit code. For example the sequence of four segments 010 111 110 110 defines a gate with A of the gate attached to pin 3 (010) of the 8 pin input of that encipher unit (numbered from pin 1 to pin 8), B attached to pin 8 (111) of the 8 pin input and C attached to pin 7 (110) of the 8 pin input. The last segment, mask M, defines that outputs A' and B' are  
20 passed through NOT gates i.e. inverted. Thus the binary values at pins 3, 7, and 8 of the input are processed by the Fredkin's gate in accordance with its internal logic as defined above and are output at pins 3, 7, and 8 of the 8 pin output coupled through that encipher unit to corresponding output pins, and the binary values at pins 1, 2, 4, 5, and 6 are coupled through that encipher unit to corresponding output pins without  
25 being processed by the Fredkin's gate. The 8 bit signal as modified by the first encipher unit is then used as an input to the second encipher unit and so on.

Figure 7 illustrates schematically an encipher apparatus having an arrangement of ten cipher units comprised of Fredkin's gates and NOT gates and Figure 8 illustrates the cipher design description used to describe the encipher apparatus.

30 As can clearly be seen the cipher unit design description simply comprises a digital code. The digital code is defined as  $((3 \times \log_2 N) + 3)$  bits and each such digital code defines a cipher unit where N is the number of input bits i.e. the data block size. The total number of bits of a cipher design description for defining an encipher apparatus

is  $M((3 \times \log_2 N) + 3)$  where  $M$  is the number of cipher units. Whilst it is possible to allow a user to select a code freely by for example choosing a "password" in ASCII code which can be translated to binary (e.g. for the 8 bit input, 10 gate example in Figure 7, a 15 character 8 bit ASCII password could be used to describe the encipher  
5 apparatus), it is preferable for the usual security reasons to randomly generate a code which describes a random configuration of the gates of the cipher units.

In this example, in order to encrypt the signal it is passed from left to right through the cipher units, i.e. from cipher unit 40a to cipher unit 40b, and so on. In order to decrypt the signal it is passed from right to left, i.e. from cipher unit 40d to cipher  
10 unit 40c, and so on, as described above. Thus in decryption as the signal is input into each cipher unit formed of a Fredkin's gate the respective mask  $M$  defines for each of the three input pins defined by the segment codes of the corresponding cipher unit design description whether the bit value on that pin is to be inverted before being operated upon by the Fredkin's gate.

15 It is possible for some of the cipher units to be implemented in parallel so long as their respective Fredkin's gate inputs are not coincident, i.e. the cipher unit design descriptions for these cipher units do not have a common 3 bit segment code.

Figure 9 is a functional diagram of a cipher unit design description generating apparatus. A random number generator 100 generates a random number to be used  
20 to form the cipher unit design description. This is input through a validity checker 110 which checks whether the random number is valid, i.e. each of the segment codes must be unique, else two or more of the gate inputs will be coupled to the same input pin. The validity checker 110 requires information on the block size in order to do this check.

25 The random number is then input into the cipher design description (circuit array) forming unit 120, constituting an encoding means of the present invention, in order to build the cipher design description describing the circuit array. The cipher design description forming unit 120 also receives an input from the cipher design description parameter selector 130, constituting a second selection means and also a third  
30 selection means of the present invention, which is operable by a user to select the number of bits or cipher units to be implemented in the cipher system, and to select the data block size. Also, in a general purpose computer, it is possible to select the type of reversible gates to be used, and in this case the computer constitutes a first

selection means of the present invention. Since there is however a limited number of possible types of gates currently known which can be implemented reversibly, allowing such a selection does not greatly increase the level of security at present.

Once the cipher design description has been formed it is then stored in a non-volatile  
5 memory 140 for use in enciphering and deciphering data. If data is to be transmitted between two parties, the cipher design description must be secretly shared. Where there has been selection of parameters in building the cipher design description i.e. the number of gates, the block size and the type of gates, this information will also need to be shared so that the cipher design description can be used properly to  
10 implement a cipher apparatus for both encryption and decryption.

Figure 10 is a functional diagram of the encipher apparatus. A signal to be enciphered is input by the data input device 200. This is then passed to a data block former 210 which forms the input signal into blocks of data which can be sequentially passed through the encipher apparatus. The first block of data is then  
15 passed into the working memory 220 as a string of N bits where N is the block size. A circuit implementor, also referred to as a cipher apparatus implementor, 250 then implements the cipher apparatus in accordance with the circuit array stored in the non-volatile memory 260. The circuit array comprises a M array of gate descriptions, where each gate description comprises four segments (as illustrated in Figure 6). The  
20 cipher apparatus implementor 250 will operate on the data block in the working memory 220 to implement each of the cipher units sequentially. Cipher apparatus implementor 250 will therefore control the data block former 410 to input a block of data into the working memory 220 when it is ready to operate on it. Once the enciphering operation has been completed on the data block in the working memory  
25 220, the cipher apparatus implementor 250 controls the passage of the data block from the working memory 220 into a memory 230. The enciphered data block can then be passed out block-by-block into a data output device 240 which can either output the enciphered data block block-by-block or can wait until all of the data blocks are enciphered and output the complete enciphered signal.

30 Figure 11 is a functional diagram of the decipher apparatus in accordance with an embodiment of the present invention. Enciphered data is received by the enciphered data input device 300 and is formed into enciphered data blocks by the enciphered data block former 310. The passage of enciphered data blocks into a working

memory 320 is then controlled by a reverse circuit implementor 350. When a data block is in the working memory 320 the reverse circuit implementor, also referred to as a reverse cipher apparatus implementor, 350 implements the decipher apparatus in accordance with the circuit array stored in the non-volatile memory 260 in reverse.

5 Once all of the cipher units defined by the circuit array have been implemented in reverse and thus the enciphered data block has been deciphered, it is output into the memory 330 under the control of the reverse cipher apparatus implementor 350. The deciphered data block can then be output to the output device 340 which can then either output each of the data blocks sequentially or wait until the complete signal  
10 has been deciphered before outputting it.

Figure 12 illustrates an application of the cipher system for the communication of enciphered data between computers 50, 51 and 52.

Computer 50 implements the encipher apparatus and generates enciphered data. This can either be stored on a non-volatile memory device such as a floppy disc 54 and  
15 passed to another computer 51 for deciphering, or it can be broadcast or transmitted over a network 53 for reception by computer 52 for deciphering there. Before the exchange of enciphered data however, it is necessary for the operators of computers 50 and 51 or 50 and 52 to secretly exchange the cipher design description. This can be done by any conventional secret means such as a secure telephone call, a secure  
20 facsimile transmission or by letter, by courier or even by a secure e-mail.

Figure 13 illustrates another embodiment of the present invention wherein a computer 60 is provided with a smartcard programmer/reader 61. In this embodiment it is possible for a smartcard to be loaded with the decipher program, i.e. software defining a software implementation of a reversible circuit for use in a  
25 cipher unit, as well as the cipher design description. The smartcard can then be given to the intended recipient of enciphered data. Thus the intended recipient of the enciphered data can simply insert the smartcard into a smartcard reader and the processor on the smartcard will implement the decipher apparatus and thus inherently the encipher apparatus. Thus the smartcard can be used for both transmitting and  
30 receiving enciphered data. This embodiment can be used by an institution such as a financial institution (Alice). A user (Bob) will be issued with a smartcard and will be able to communicate securely with the institution by inserting the smartcard into the reader 61 e.g. an automatic teller machine (ATM).

Figure 14 is a schematic diagram of the implementation of the cipher apparatus in a general purpose computer. The computer is provided with a bus 79 to communicate between operational units. A modem 70 is provided for connection over a telecommunications line 78 to transmit and receive enciphered data. Also a network card 80 is provided for connection over a network to transmit and receive data. A keyboard 74 is provided for inputting data and a display 71 is provided for displaying deciphered data. A processor 72 implements the cipher apparatus either in a forward direction for enciphering or in a reverse direction for deciphering in accordance with the circuit array stored in the memory section 77. The processor 72 operates in accordance with the circuit emulation program stored in the program memory 75. During the operation of the processor 72 data is temporarily stored in the working memory 76 and at the end of the enciphering or deciphering process the enciphered or deciphered data can be stored in the data storage device 73 which can comprise non-volatile storage media such as a floppy disc, a hard disc, a writable CD-ROM, or EPROM.

The method of operation of the cipher apparatus of this embodiment of the present invention will now be described with reference to Figures 15 to 18.

Figure 15 illustrates the steps involved in the generation of the cipher design description, enciphering of data, the transmission of the enciphered data and the deciphering of the enciphered data.

In step S1 a type of reversible processing is predetermined or selected e.g. Fredkin's gates. In step S2 the number of gates M and the block size N of the data is selected. In step S3 the cipher design description (or circuit array) is then generated. In step S4 the cipher design description is exchanged secretly between Alice and Bob. In step S5 Alice enciphers data using an cipher apparatus configured according to the circuit array, i.e. as an encipher apparatus. Alice then communicates the enciphered data to Bob in step S6. In step S7 Bob decipheres the enciphered data using a reverse cipher apparatus, i.e. a decipher apparatus, configured appropriately according to the circuit array (cipher design description).

Figure 16a illustrates in more detail the steps involved in a generation of the cipher design description.

In step S10 a variable m is set to 0. This variable acts as the cipher unit (gate) number. In step S11 a random number having P bits is then generated, where P is

the number of bits necessary to describe a cipher unit. In the example given hereinabove using Fredkin's gates,  $P=12$  (4 segments each of 3 bits). In step S12 a check is carried out to determine whether this is a valid random number. One of the tests is whether the random number defines the cipher unit having two or more pins  
5 on the same input data address which is not allowed. In step S13 if the random number is valid the cipher unit number  $m$  is incremented and in step S14 the generated random number is stored indexed by  $m$ . In step S15 it is then determined whether random numbers have been generated for all of the cipher units i.e.  $m=M$ . If not, the process returns to step S11 for the generation of further random numbers.  
( 10 If random numbers have been generated defining all of the cipher units then the process ends in step S16.

Figure 16b illustrates the circuit array which comprises a  $P \times M$  matrix. The matrix is indexed by  $M$  where each entry comprises  $P$  bits divided into four segments A, B, C and M each of 3 bits.

15 The process of enciphering data will now be described with reference to Figure 17. In step S20 the data to be enciphered is input and the prestored circuit array (secret cipher design description) indexed by  $m$  is read. The first  $N$  bits of data are then read as a data block. If there are less than  $N$  bits of data, padding data is generated in order to make up  $N$  bits. The  $N$  bits of data are then loaded into the working array in  
20 step S22 and in step S23 the cipher unit counter  $m$  is set to 1. In step S24 the first segment A for cipher unit  $m$  in the circuit array is read and this is used to address a data bit from the working array in step S25. In step S26 it is then determined whether the read data bit is 0. If it is not 0 then there is no exchange of data between input B and C and the process proceeds to step S30. If it is 0 then data bits  
( 25 B and C are exchanged. Thus in step S27 the second and third segments B and C for the cipher unit  $m$  in the circuit array are then read and these are used to address two data bits in the working array. In step S29 these data bits are then exchanged and the process proceeds to step S30 where a mask bit counter  $b$  is set to 1 to index the first mask bit for segment A.  
30 In step S31 the  $b^{\text{th}}$  bit of the mask is read and in step S32 it is determined whether this is zero. If it is not zero the data bit in the working array addressed by the  $b^{\text{th}}$  segment is inverted in step S33 otherwise no action is taken. In the next step S34 is determined whether all of the mask bits have been read i.e.  $b=3$  indicating that the



mask bit for segment C has been read. If not the mask bit counter  $b$  is incremented in step S35 to index the next mask bit for segment B or C and the process returns to step S31. If all of the mask bits have been read it is then determined whether all of the cipher units have been implemented i.e.  $m = M$  in step S36. If not, the cipher unit  
5 counter  $m$  is incremented and the process returns to step S24. Otherwise in step S38 the working array is output as a block of enciphered data. In step S39 it is then determined whether the data has all been enciphered and if not, in step S40 the next  $N$  bits of data are input and padded if necessary and the process returns to step S22. Otherwise the process ends in step S41 since all of the data has been enciphered.

- ( 10 The process of deciphering enciphered data will now be described with reference to Figure 18.

In step S50 the enciphered data is input and the prestored circuit array indexed by  $m$  is read. The first  $N$  bits of enciphered data are then read in step S51. The  $N$  bits of enciphered data are then loaded into the working array in step S52 and in step S53  
15 the cipher unit counter  $m$  is set equal to  $M$  i.e. the first cipher unit to be implemented is in fact the last cipher unit in the array so that the cipher unit are implemented sequentially in reverse. In step S54 the mask bit counter  $b$  is then set to the first mask bit for segment A and in step S55 the  $b^{\text{th}}$  bit of the mask is read. It is then checked in step S56 whether this is zero and if not the data bit in the working array  
20 addressed by the  $b^{\text{th}}$  segment is inverted in step S57 otherwise no action is taken. The process then proceeds to step S58 wherein it is determined whether all of the mask bits have been read i.e.  $b = 3$ . If not, in step S59 the mask bit counter  $b$  is incremented to index the next mask bit for segment B or C and the process returns to step S55. If all of the mask bits have been read for the mask segment, in step S60  
25 the first segment A for cipher unit  $m$  in the circuit array is read. A data bit in the working array addressed by this first segment A is then read in step S61 and it is determined whether this is zero in step S62. If it is zero the second and third segments B and C for cipher unit  $m$  in the circuit array are read in step S63 and in step S64 the data bits in the working array addressed by the second and third  
30 segments B and C are read. These are then exchanged in step S65 and the process proceeds to step S66. If in step S62 the data bit addressed by the first segment A is not zero the process proceeds to step S66. In step S66 it is determined whether the process has just been carried out for the first cipher unit i.e.  $m = 1$  indicating that the

deciphering of the current block has finished. If not in step S67 the cipher unit counter is decremented and the process returns to step S54, and if so in step S68 the working array is output as a deciphered data block. In step S69 it is determined whether all of the blocks have been deciphered and if not in step S70 the next N bits  
5 of deciphered data are read. The process then returns to step S52. If in step S69 it is determined that all of the data has been deciphered, in step S71 any data corresponding to padding data in the last block is ignored and the process ends in step S72.

( In the above enciphering and deciphering embodiment each cipher unit is  
10 implemented in software sequentially.

In the embodiment described hereinabove the reversible circuit (cipher unit) is implemented by a Fredkin's gate. The Fredkin's gate can be viewed as a three-input, three-output logic gate as illustrated in Figure 19. This can be implemented using AND, OR and NOT logic gates (which are not reversible) as illustrated in Figure 20.  
15 Of course, since the logic gates can only conduct signals one-way in order for the circuit to be reversible, it must perform an operation which is symmetric i.e. if the output of the circuit is put back as an input, the original input will be obtained. This is because a Fredkin's gate is an inverse of itself.

Thus the circuit illustrated in Figure 20 can form the basis of a cipher unit 40 of an  
20 encipher apparatus or a decipher apparatus (Figures 3 and 4).

( Another implementation of the Fredkin's gate can be chosen using multiplexers as illustrated in Figure 21. Each multiplexer 400 and 401 receives two input signals and one control signal. If the control signal is zero then the first input signal is passed. If the control signal is one the second input signal is passed.

25 The Fredkin's gate can also be implemented by three-state buses as illustrated in Figure 22.

All of the three circuits given hereinabove can be implemented either in software using a computer program which generates the circuits and simulates them or using electronic hardwired circuits. It is thus possible for Alice and Bob to be supplied with  
30 off the shelf programmable logic gate array (PLGA) chips and with a storage medium storing logic to configure the PLGA to carry out the encipher method or the decipher method of the present invention, i.e. the software that downloads the circuit description onto the chip. Such software languages for circuit descriptions can for

example comprise Verilog-HDL. In order for Alice and Bob to establish the secret communications, the downloading of the cipher circuit description will only be done once. When the circuit is implemented using hardwired circuits, Alice and Bob can use one circuit for encryption and one circuit for decryption. It is however possible  
 5 to use only one circuit by downloading the circuit description at the time when communications take place. For example, if Alice wishes to send an encrypted message she downloads the cipher design description (circuit array) onto the chip. If she receives a message she can download the corresponding decryption circuit (its circuit description) onto the chip.

( 10 In the embodiment given above, the Fredkin's gate is implemented in logic. However, a Fredkin's gate can be implemented in many different ways, for example, it is possible to implement the Fredkin's gate in optics. The device which can be used to implement the Fredkin's gate in optics is the Mach/Zehnder interferometer switch. Such a switch is disclosed in a paper by J. Dommelly *et al* entitled "A Gallium  
 15 Arsenide Electro-optical Interferometer Modulator", (Proc. 7th Topical Meeting on Integrated and Guided Wave Optics, Kissimmee 1984).

Although in the above embodiments, the use of Fredkin's gate has been described, the cipher units of the present invention can be implemented in many different ways. For example, another form of reversible universal logic is the AND/NAND gate (which  
 20 is also known as Toffoli's gate). The operation of the AND/NAND gate can be given by:

$$\begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_{n-1} \\ x_n \end{pmatrix} \rightarrow \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_{n-1} \\ x_n \oplus x_1 x_2 \dots x_{n-1} \end{pmatrix}$$

The AND/NAND gate is illustrated in Figure 23. In this gate the input on (n-1) of the  
 25 n inputs act to switch the n<sup>th</sup> input by virtue of an AND gate receiving the (n-1) inputs and acting on an XOR gate on the n<sup>th</sup> input. Details on this particular type of gate are given in the paper by T. Toffoli entitled "Bicontinuous Extensions of Invertible Combinatorial Functions" (Mathematical Systems Theory, Vol. 14, pp. 13-23).

The AND/NAND gate can be implemented not just in logic as illustrated in Figure 23, but by any physical system.

In a system for implementing the cipher units of the present invention, any reversible computational system can be used and the present invention is not limited to the use of circuit implementation. For example, reversible cellular automata can be used as described in "Computation and Construction Universality of Reversible Cellular Automata" by T. Toffoli (J. Comput. Sys. Sci., Vol. 15, 1977, pp. 213-231), a reversible Turing machine as described in "Logical Reversibility of Computation" by C. Bennett (IBM J. Res. Dev. 6, 1973 pp. 525-532), quantum computing, for a "billiard ball", model of computation as described in "Conservative Logic" by E. Fredkin and T. Toffoli (International Journal of Theoretical Physics, Vol 21. nos. 3/4, 1982), for example.

In the embodiments described hereinabove, for security, a random number generator is used in order to randomly generate a circuit configuration. The random number generator is not essential to the present invention but does increase the level of security. Any of the standard strong real number generators available for crypto-software libraries can be used, or a true physical random source can be used. The generation of random or pseudo-random numbers is well known in the art.

It will be apparent to the skilled person in the art that the present invention can be implemented by providing Alice and Bob with a program that randomly generates circuits and simulates them. Generally in the software implementation the circuits will only be generated and simulated using the cipher design description (circuit array) when the signal is input to be enciphered or deciphered.

If the cipher apparatus is to be implemented using programmable hardware, a manufacturer will provide Alice and Bob with a conventional programmable logic gate array and a storage medium storing logic to set it up to run as a cipher apparatus.

## CLAIMS:

1. Encipher apparatus for enciphering a signal, comprising:  
forming means for receiving the signal to be enciphered and for outputting  
5 the signal as a sequence of data blocks, each having a first predetermined number of bits;  
a plurality of encipher functional modules sequentially coupled to operate sequentially on the sequence of data blocks from the forming means; and  
configuring means,  
10 wherein each encipher functional module comprises  
a module input,  
a module output, and  
a respective data processing unit having a parallel input and a corresponding parallel output and being arranged to perform a respective reversible process upon a set of  
15 bits at its parallel input and to produce at its corresponding parallel output a corresponding enciphered set of bits,  
and is operable under the control of the configuring means to couple a respective predetermined set of the bits of a data block received at its module input to the parallel input of its data processing unit and to provide at its module output an  
20 enciphered data block in which said respective predetermined set of bits is replaced by the corresponding enciphered set of bits produced at the parallel output of its data processing unit.
2. Encipher apparatus according to claim 1, wherein said respective data  
25 processing units are of a single type.
3. Encipher apparatus according to claim 1 or claim 2, wherein the reversible process of at least one of said data processing units is a switching operation controlled by at least one of the set of bits received at its parallel input.  
30
4. Encipher apparatus according to any preceding claim wherein each of said data processing units is a reversible gate.

5. Encipher apparatus according to claim 3, wherein said reversible gate comprises a Fredkin's gate or an AND/NAND gate.
6. Encipher apparatus according to any preceding claim wherein said  
5 configuring means is operative to control said encipher functional modules in accordance with a cipher design description.
7. Encipher apparatus according to claim 6, including means for receiving said cipher design description.
- 10 8. Encipher apparatus according to claim 6, including means for generating said cipher design description.
9. Encipher apparatus according to claim 8, wherein the generating means  
15 includes a random or pseudo-random number generator and is operative to use random or pseudo-random numbers generated by said random or pseudo-random number generator to describe in code said respective predetermined sets of bits.
10. Encipher apparatus according to any preceding claim, wherein each said  
20 encipher functional module comprises a logic gate which does not conserve logic.
11. Encipher apparatus according to any preceding claim, wherein said plurality of encipher functional modules form a programmable circuit.
- 25 12. Encipher apparatus according to claim 11, wherein said plurality of encipher functional modules comprise a programmable logic gate array, and said configuring means comprises a programming means for programming said programmable logic gate array.
- 30 13. Encipher apparatus according to claim 11, wherein said encipher functional modules comprise analogue electronic modules.

14. Encipher apparatus according to any one of claims 1 to 10, wherein said signal is an optical signal and said encipher functional modules comprise optical components.
- 5 15. Encipher apparatus according to any one of claims 1 to 10, comprising a programmable computing apparatus, wherein said encipher functional modules comprise a computer code routine implemented on said programmable computing apparatus.
- 10 16. Encipher apparatus according to claim 15, wherein said computer code routine is in the form of a generic module code routine repeatedly implemented dependent upon information from said configuring means.
17. Encipher apparatus according to any preceding claim including first selection  
15 means for selecting a type of encipher functional module to be used from amongst a plurality of possible types of encipher functional modules, wherein said configuring means is adapted to configure the encipher apparatus to use the selected type of encipher functional module.
- 20 18. Encipher apparatus according to any preceding claim, including second selection means for selecting the number of said encipher functional modules to be used, wherein said configuring means is adapted to configure the encipher apparatus to use the selected number of encipher functional modules.
- 25 19. Encipher apparatus according to any preceding claim including third selection means for selecting for each said encipher functional module the respective predetermined set of the bits of a data block received at its module input.
20. A method of enciphering a signal, the method comprising:  
30 receiving the signal to be enciphered and forming the signal into a sequence of data blocks, each having a first predetermined number of bits;

applying the sequence of data blocks to a plurality of encipher functional modules sequentially coupled to operate sequentially on the sequence of data blocks, each encipher functional module comprising

- a module input,
- 5 a module output, and
- a respective data processing unit having a parallel input and a corresponding parallel output and being arranged to perform a respective reversible process upon a set of bits at its parallel input and to produce at its corresponding parallel output a corresponding enciphered set of bits; and
- 10 configuring each encipher functional module to couple a respective predetermined set of the bits of a data block received at its module input to the parallel input of its data processing unit and to provide at its module output an enciphered data block in which said respective predetermined set of bits is replaced by the corresponding enciphered set of bits produced at the parallel output of its data
- 15 processing unit.

21. A method according to claim 20, wherein the encipher functional modules are of a single type.

- 20 22. A method according to claim 20 or claim 21, wherein the reversible process of at least one of said data processing units is a switching operation controlled by at least one of the bits of a data block received at its parallel input.

23. A method according to any one of claims 20 to 22, wherein said encipher  
25 functional modules each act as a reversible gate.

24. A method according to any one of claims 20 to 23, wherein the configuring of said encipher functional modules is in accordance with a cipher design description.

- 30 25. A method according to claim 24, including receiving said cipher design description.



26. A method according to claim 24, including generating said cipher design description.

27. A method according to claim 24, including generating random or pseudo-  
5 random numbers and using the generated random or pseudo-random numbers to generate said cipher design description.

28. A method according to claim 27, wherein a respective generated random or pseudo-random number is used to described in code the respective predetermined set  
10 of bits for a respective said encipher functional module.

29. A method according to claim 28, wherein the logic operations do not conserve logic.

15 30. A method according to any one of claims 20 to 29, wherein said encipher functional modules comprise a programmable logic gate array and the configuring step includes programming said programmable logic gate array.

31. A method according to any one of claims 20 to 29, implemented by  
20 computer code on a computing apparatus, wherein said encipher functional modules comprise a computer code routine implemented in dependence upon configuration information.

32. A method according to claim 31, wherein the computer code routine is  
25 implemented repeatedly dependent upon the number of said encipher functional modules to be implemented.

33. A method according to any one of claims 20 to 32, including selecting the type of encipher functional module to be used from amongst a plurality of possible  
30 types of encipher functional modules.

34. A method according to any one of claims 20 to 33, including selecting the number of said encipher functional modules used.

35. A method according to any one of claims 20 to 34, including selecting the respective predetermined set of the bits of a received data block for said encipher functional modules.

5

36. Decipher apparatus for deciphering a signal, comprising:  
forming means for receiving the signal to be deciphered and for outputting the signal as a sequence of data blocks, each having a first predetermined number of bits;

10 a plurality of decipher functional modules sequentially coupled to operate sequentially on the sequence of data blocks from the forming means; and  
configuring means,

wherein each decipher functional module comprises

a module input,

15

a module output, and

a respective data processing unit having a parallel input and a corresponding parallel output and being arranged to perform a respective reversible process upon a set of bits at its parallel input and to produce at its corresponding parallel output a corresponding enciphered set of bits,

20 and is operable under the control of the configuring means to couple a respective predetermined set of the bits of a data block received at its module input to the parallel input of its data processing unit and to provide at its module output a deciphered data block in which said respective predetermined set of bits is replaced by the corresponding deciphered set of bits produced at the parallel output of its data  
25 processing unit.

37. Decipher apparatus to claim 36, wherein said decipher functional modules are of a single type.

30 38. Decipher apparatus according to claim 36 or claim 37, wherein said configuring means is operative to control said decipher functional modules in accordance with a cipher design description.

39. Decipher apparatus according to claim 38, wherein said cipher design description is equivalent to the inverse of a cipher design description used to control encipher functional modules of an encipher apparatus used to produce the enciphered signal.

5

40. Decipher apparatus according to claim 38 or claim 39, including means for receiving said cipher design description.

41. Decipher apparatus according to claim 38 or claim 39, including means for  
10 generating said cipher design description.

42. Decipher apparatus according to claim 41, wherein the generating means includes a random or pseudo-random number generator and is operative to use random or pseudo-random numbers generated by said random or pseudo-random  
15 number generator to describe in code said respective predetermined sets of bits.

43. Decipher apparatus according to any one of claims 36 to 42, wherein the reversible process of at least one of said data processing units is a switching operation controlled by at least one of the bits of a data block received at its parallel  
20 input.

44. Decipher apparatus according to any one of claims 36 to 43, wherein each of said data processing units comprises a reversible gate.

25 45. Decipher apparatus according to claim 44, wherein said reversible gate comprises a Fredkin's gate or an AND/NAND gate.

46. Decipher apparatus according to any one of claims 36 to 45, wherein each said decipher functional module comprises a logic gate which does not conserve  
30 logic.

47. Decipher apparatus according to any one of claims 36 to 46, wherein said plurality of decipher functional modules form a programmable circuit.

48. Decipher apparatus according to claim 47, wherein said plurality of decipher functional modules comprise a programmable logic gate array, and said configuring means comprises a programming means for programming said programmable logic gate array.

49. Decipher apparatus according to any one of claims 36 to 46, wherein said signal is an optical signal and said decipher functional modules comprise optical components.

10

50. Decipher apparatus according to any one of claims 36 to 46, comprising a programmable computing apparatus, wherein said decipher functional modules comprise a computer code routine implemented on said programmable computing apparatus.

15

51. Decipher apparatus according to claim 50, wherein said decipher functional modules comprise a computer code routine repeatedly implemented dependent upon information from said configuring means.

20 52. Decipher apparatus according to any one of claims 36 to 51, wherein said configuring means is responsive to type identifying information included in a cipher design description to configure the type of said decipher functional modules in accordance with said type identifying information.

25 53. Decipher apparatus according to any one of claims 36 to 52, wherein said configuring means is responsive to module number information included in a cipher design description to configure a corresponding number of said decipher functional modules.

30 54. Decipher apparatus according to any one of claims 36 to 53, wherein said configuring means is responsive to data block size information included in a cipher design description adapted to configure the input and output of each said decipher functional module.

55. A method of deciphering an enciphered signal, the method comprising:  
receiving the signal to be deciphered and outputting the signal as a sequence  
of data blocks, each having a first predetermined number of bits;
- 5 applying the sequence of data blocks to a plurality of decipher functional  
modules sequentially coupled to operate sequentially on the sequence of data blocks,  
each decipher functional module comprising  
a module input,  
a module output, and
- 10 a respective data processing unit having a parallel input and a corresponding parallel  
output and being arranged to perform a respective reversible process upon a set of  
bits at its parallel input and to produce at its corresponding parallel output a  
corresponding enciphered set of bits; and  
configuring each decipher functional module to couple a respective
- 15 predetermined set of the bits of a data block received at its module input to the  
parallel input of its data processing unit and to provide at its module output an  
deciphered data block in which said respective predetermined set of bits is replaced  
by the corresponding deciphered set of bits produced at the parallel output of its data  
processing unit.
- 20
56. A method according to claim 55, wherein the decipher functional modules  
are of a single type.
57. A method according to claim 55 or claim 56, wherein the reversible process
- 25 of at least one of said data processing units is a switching operation controlled by at  
least one of the bits of a data block received at its parallel input.
58. A method according to any one of claims 55 to 57, wherein said decipher  
functional modules each act as a reversible gate.
- 30
59. A method according to any one of claims 55 to 58, wherein the configuring  
of said decipher functional modules is in accordance with a cipher design description.

60. A method according to claim 59, including receiving said cipher design description.

61. A method according to claim 59, including generating said cipher design  
5 description.

62. A method according to claim 59, including generating random or pseudo-random numbers and using the generated random or pseudo-random numbers to generate said cipher design description.

10

63. A method according to claim 62, wherein a respective generated random or pseudo-random number is used to described in code the respective predetermined set of bits for a respective said decipher functional module.

15 64. A method according to claim 63, wherein the logic operations do not conserve logic.

65. A method according to any one of claims 55 to 64, wherein said decipher functional modules comprise a programmable logic gate array and the configuring  
20 step includes programming said programmable logic gate array.

66. A method according to any one of claims 55 to 64, implemented by computer code on a computing apparatus, wherein said decipher functional modules comprise a computer code routine implemented in dependence upon configuration  
25 information.

67. A method according to claim 66, wherein the computer code routine is implemented repeatedly dependent upon the number of said decipher functional modules to be implemented.

30

68. A method according to any one of claims 55 to 67, including selecting the type of decipher functional module to be used from amongst a plurality of possible types of decipher functional modules.

69. A method according to any one of claims 55 to 68, including selecting the number of said decipher functional modules used.
- 5 70. A method according to any one of claims 55 to 69, including selecting the respective predetermined set of the bits of a received data block for said decipher functional modules.
71. Apparatus for generating a cipher design description, comprising:  
10 a random or pseudo-random number generator; and  
encoding means arranged to receive a random or pseudo-random number generated by said generator and to encode that received number to provide at its output a cipher design description describing, for each of a plurality of cipher functional modules sequentially coupled to operate sequentially on a data block  
15 applied to the plurality of sequentially coupled cipher functional modules, a respective predetermined set of the bits of the data block as received at the respective module's input.
72. Apparatus according to claim 61, including first selection means for selecting  
20 at least one type of cipher functional module to be used from amongst a plurality of possible types of cipher functional modules, said encoding means being adapted to include the selected type or types in the encoded information.
73. Apparatus according to claim 71 or claim 72, including second selection  
25 means for selecting the number of said cipher functional modules to be used, said encoding means being adapted to include the selected number in the encoded information.
74. Apparatus according to any one of claims 71 to 73, including third selection  
30 means for selecting the number of inputs and outputs of said cipher functional modules, said encoding means being adapted to include the selected number in the encoded information.

75. A method of generating a cipher design description, the method comprising:  
generating random or pseudo-random numbers;  
encoding information, including said random or pseudo-random numbers, describing,  
for each of a plurality of cipher functional modules sequentially coupled to operate  
5 sequentially on a data block applied to the plurality of sequentially coupled cipher  
functional modules, a respective predetermined set of the bits of the data block as  
received at the respective module's input; and  
outputting the cipher design description.
- 10 76. A method according to claim 75, further comprising selecting at least one  
type of cipher functional module to be used from amongst a plurality of possible  
types of cipher functional modules, and including the selected type or types in the  
encoded information.
- 15 77. A method according to claim 75 or claim 76, further comprising selecting the  
number of said cipher functional modules used, and including in the encoded  
information said selected number of said cipher functional modules used.
78. A method according to any one of claims 75 to 77, further comprising  
20 selecting the number of inputs and outputs of said cipher functional modules, and  
including in the encoded information the selected number of inputs and outputs of  
said cipher functional modules.
79. Cipher apparatus comprising the encipher apparatus of any one of claims 1  
25 to 19 and the decipher apparatus of any one of claims 36 to 54, wherein the  
encipher functional modules of the encipher apparatus are constituted by the  
decipher functional modules of the decipher apparatus but are sequentially coupled in  
the opposite order.
- 30 80. A cipher method for enciphering and deciphering a signal comprising the  
encipher method of any one of claims 20 to 35 and the decipher method of any one  
of claims 55 to 70.



81. Processor implementable instructions for controlling a processor to carry out the method of any one of claims 20 to 35, 55 to 70, 75 to 78, and 80.

82. A carrier medium carrying the processor implementable instructions  
5 according to claim 81.

83. A storage medium storing logic to configure a programmable logic gate array to carry out the method of any one of claims 20 to 35, 55 to 70, 75 to 78, and 80.